2003

Automatic localization of craniofacial landmarks of cephalograms using artificial neural networks.

Idris S. El-Feghi
University of Windsor

AUTOMATIC LOCALIZATION OF CRANIOFACIAL LANDMARKS OF CEPHALOGRAMS USING ARTIFICIAL NEURAL NETWORKS

By

Idris S. El-Feghi

A Dissertation
Submitted to the Faculty of Graduate Studies and Research
through Electrical and Computer Engineering
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada

2003

Abstract

In modern orthodontic practice, a great reliance is placed on objective and systematic methods of characterizing craniofacial forms, using measurements based on a set of agreed upon points known as craniofacial landmarks. Lateral skull x-ray images are usually used in cephalometric analysis to provide quantitative measurements of the head. Accurate location of those landmarks forms the basis for what is known as cephalometric evaluation. Distance and angles among these landmarks are compared with normative values to diagnose a patient's deviations from ideal form, evaluate the craniofacial growth and measure the effect of the treatment.

Because of the large variability in the morphology of the human head, large variations of special coordinates of landmarks are observed and must be reduced. To reduce this variation, adaptive localization based on the size, rotation and shifts of the skull is used. The adaptive system requires a training set that will account for all the variations in the cephalograms. A good training set is difficult to obtain due to unavailability of fixed workbenches of locations of landmarks that cephalometric measurements of x-rays can be compared with. To create a reliable training set, images are grouped into several clusters and one prototype representing that cluster is used in the training set.

The work in this thesis reports two novel algorithms for locating craniofacial landmarks on digitized skull x-rays. The first algorithm is based on the use of neuro-fuzzy networks to minimize the search windows for each landmark. Parametric template matching is then used to pin point the exact location of the landmark inside a search window. The second algorithm uses a Multi-Layer Perceptron as a function approximator to predict the location of the landmark based on learned knowledge obtained from a training set.

A new method for extracting a features vector from each image is also reported. This feature vector is used to represent images and also used for clustering images to obtain a reliable training set using K-means after providing it with initial estimates of centers of the groups. To reduce the dimension of the feature vector, we provide an efficient pruning technique for reduction of features based on sensitivity analysis. It is shown that this reduction will minimize the number of rules required for the fuzzy system while the clustering characteristics are preserved.

Algorithms are simulated using C++ code. Results obtained using the two algorithms are compared with previous works. It is shown that the proposed algorithms outperform other methods found in the open literature.

To my wife Fuzia and to my children Khiria, Mohamed and Malik who endured the hardship of this work more than I.

Acknowledgments

I would like to express my sincere thanks and appreciation to my co-supervisors, Dr. M. A. Sid-Ahmed and Dr. M. Ahmadi for their invaluable advices, guidance, and constant encouragement and support through the progress of this thesis. I would like also to thank my committee members Dr. M. Omair Ahmad, Dr. B. Boufama, Dr. C. Chen and Dr. X. Chen for their time and advices.

I am thankful to Dr. S. Erfani for his encouragements and his critical comments and to Ian Mitchell for proofreading this thesis.

Finally, I would like also to thank my friends and colleagues in the Signal Processing Lab. for happy memories we had during my work on this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Introduction

Analysis of cephalogram images and measurement of parameters based on detected feature points referred to as landmarks, play an essential role in diagnosis and treatment planning by orthodontists. In order to provide measurements of the craniofacial skeleton, there exists a set of agreed-upon feature points reference. The number of landmarks that have been named and defined is quite large. Admittedly, many of these landmarks are rarely used. Rakosi [1] defines a smaller set of 20-30 landmarks, which contains those most frequently used by orthodontists in what is called cephalometric evaluation.

Lines are constructed from the detected landmarks, and the subsequent measurement of various angular and linear parameters generates an evaluation of the facial skeleton and teeth for the orthodontist to assess. The conventional method of locating landmarks depends on manual tracing of radiographic images to locate landmarks on line crossing. This can take an experienced orthodontist up to 20 minutes. The process is tedious, time consuming and subject to

human error. An automated system would not only eliminate the above-mentioned problems and produce repeatable results but would also be useful for compiling dental records for fast retrieval.

1.2 Previous Work

Investigations into automatic landmark recognition using different image processing techniques have been undertaken for several years. The task is a difficult one due to the complexity and variability of the images. Automatic cephalometric analysis has been subject to research for many years and the task of automatic landmarking has been attempted by more than 20 independent researchers with varying degrees of success [2-21]. The research can be categorized by three methodologies. The first category employed knowledge-based edge tracker driven by a predetermined model. In the model-driven approaches, edge information from X-ray images are extracted so that desired features can be found via line trackers in a predetermined order. Another method searches for known patterns in the images. It can use various well-developed techniques but requires pre-definition of patterns or systematic training. The second category followed a strategy similar to that used by orthodontists. This approach employed a combination of image processing techniques to extract important edges, which would then be used to locate or estimate the location of the landmarks on line crossings. It used geometrical means to reduce the search area, then utilized image-matching techniques to pinpoint the exact location of landmarks. The third category employed artificial intelligence and neural networks.

It was Levy-Mandel *et al, (1986)* [2], who was the first to attempt to automate the location of landmarks using knowledge-based line extraction techniques. Their system consisted of three stages. The first stage is a prefiltering stage, in which a median filter is used, followed by histogram equalization, and an image sharpening technique to enhance the contrast and remove the noise. In the

second stage a Mero-Vassy Operator for edge detection is applied to extract the relevant edges using a knowledge-based line tracker. Lines are tracked in a predetermined order based on algorithms introduced to the system by ad hoc criteria. In the third stage landmarks are then located on line crossing according to their geometric definitions. The system was tested on two high quality x-rays; the positions of 23 out of 36 landmarks were determined. This system can only detect landmarks located on or near edges and its performance relies on the quality of the x-rays.

Yen *et al, (1986)* [3], presented an extension to the work of Levy-Mandel [2] by providing a more efficient algorithm for performing the median filtering and better representation of the knowledge base. They also developed 3-D Landmarking techniques for achieving better results. The system was tested on three cephalograms. The system located 21 landmarks on the first x-ray, 13 on the second and 10 on the third.

Parthasarathy *et al.* [4] in1989 improved the system used in [3] by including a four-level resolution pyramid to reduce the processing time. They used 480x512 pixel images then the resolution of x-ray images is reduced to 64x60 pixels, once the landmarks are found the resolution is scaled back to the original size and the locations are fine-tuned. The system was tested on five cephalograms of varying quality to locate nine landmarks. On average 18 percent were located within 1mm, 58 percent within 2mm and 100 percent within 5mm.

Tong *et al. [5] in* 1990 presented an extension to the work of [4]. Their system tried to locate 17 landmarks, not including any of the seven of the earlier work. They indicated that their results if combined with [3] would yield a full cephalometric analysis. If both systems were combined together, as was intended on the five cephalograms, on average 40 percent of 26 landmarks would be located to within 1 mm, 70 percent to within 2 mm and 95 percent to within 5 mm.

Davis and Taylor [6] in 1991 described how hand-crafted algorithms could be integrated in a blackboard architecture, allowing back-tracking in the face of contradictions. A formal evaluation by Forsyth and Davis [7] in 1996 used 10 cephalograms scanned at 512 x 512 pixels with 64 gray levels. It excluded radiographs, where erupted teeth were overlying the apices of the incisors. On average 63 percent of 19 landmarks were located to within 1 mm and 74 percent to within 2 mm.

Contreras *et al* [8], Jackson *et al* [9], Cohen *et al* [10-11] and Davis [12] presented similar knowledge-based edge tracking methods. The approaches given in [2-11] suffered from similar problems due to the use of rigid knowledge-based rules. The main disadvantage of edge tracking systems, is that landmarks found by edge detection are very sensitive to poor quality x-rays. Another disadvantage of knowledge-based approaches is the difficulty of adding a new landmark to the system, for which new rules have to be formulated and added to the rule base. In these earlier works the line-tracking algorithms were tested using the same images used to create them. The algorithms were never tested on unseen images.

The second category of researchers used mathematical or statistical models to narrow down the search area for each landmark, then used template-matching techniques to pin point the exact location of landmarks. Cardillo and Sid-Ahmed [13] presented the most promising results locating 76% of the landmarks within 2mm. They used a combination of mathematical modeling methods similar to affine transformation to reduce the size of the search window. They then applied a shape recognition algorithm based on gray-scale mathematical morphology to exactly locate the landmark. Decomposition of shapes was used to desensitize the algorithm to size difference. Although this method produced good results it is based on a small set of approximately 40 x-rays.

Grau *et al* [14] tried to improve the work of Cardillo [12] by using a line detection module to search for the most significant lines, such as the jaw line or nasal

spine, then utilized the same mathematical morphology developed in [13] for shape recognition. They used only one component for shape matching, which meant that the algorithm was not desensitized to size differences. They used 20 images for training and testing the algorithm. Their methods produced some improvements over [13] using a smaller image set for testing of approximately 20 x-rays.

Desvignes *et al* [15] addressed the problem of finding an initial estimate of the location of landmarks using adaptive coordinate space where locations are registered. To reduce the variability and to produce an initial estimate of the region of interest, they used statistical models and training sets. The algorithm was tested on a set of 58 x-rays of which 28 were used for training. Although they claimed to locate 99.5% of the landmarks within a window, the mean error between the real position and the estimated position was about 4.5 mm. This error was significantly large compared to the permissible error of ±2 mm.

Hutton et *al* [16] used active shape models for cephalometric landmarking. Permissible deformations of a template were established from a training set of hand-annotated images and the resulting model was used to fit unseen images. Sixty-three randomly selected cephalograms were tested using a drop-one-out method. On average, 13 per cent of 16 landmarks were within 1 mm, 35 per cent were within 2 mm, and 74 per cent were within 5 mm. It was concluded that the current implementation did not give sufficient accuracy for complete automated landmarking, but could be used as a timesaving tool to provide a first-estimate location of the landmarks.

Rudolph *et al.* [17] described the use of spatial spectroscopy to characterize the gray-level appearance around each landmark from a training set of hand laid labeled images. They used a "drop-one-out" scheme to enable the testing of 14 images using the other 13 as the training set. However, they used images that were just 64 x 64 pixels to make their scheme computationally feasible. They

compared their system's performance with that of an expert, using images of the same resolution. They implied that 100 percent of the landmarks were located to within 4 mm. They suggested that as the resolution increased, the landmarks would correspondingly become more accurate.

The third category of researchers used neural networks and fuzzy inference systems to locate the landmarks. Neural networks can be trained to locate the landmarks on a training set to locate or estimate the locations on target images. Fuzzy neural networks offer a more flexible solution because they formulate a solution based on fuzzy rules. Uchino *et al* [18] presented a fuzzy learning machine that could learn the relation between the mean gray levels of the image and the location of the landmarks. They divided the image into nine blocks where the mean of the gray level of each block constituted an input to the machine. The weights on the consequence were adjusted during the learning stage using the gradient decent algorithm. Once the machine located the block containing the landmark, it was divided into 9 separate blocks. The process is repeated until the size of the block is reduced and exact landmarks are located. This method only works if images are registered in a common space, having the same size rotation and shift. This method suffers from several drawbacks such as the training time, the amount of weights used and the number of fuzzy rules. Each landmark will require nine levels of training to produce nine sets of weights. The number of rules is equal to (number of sets)^9. The lowest number of sets that can be used in the output is two, making the number of rules equal to 512. The number of weights is equal to 9x512. This method was tested on images not used for training producing an error of ±2.3mm, which was close to the allowed error of ±2mm.

Sanei *et al.*[19] proposed an iterative self-organized data analysis technique using a combination of statistical pattern recognition and fuzzy logic-based decision-making technique for detection of craniofacial landmarks on x-ray images of size 256x256. The images were primarily processed in order to enhance the regions of soft and hard tissues separately. Then for each landmark,

different template matching plus fuzzy logic approach has been followed. Final landmarks are selected from a number of detected points by applying another fuzzy decision-making criterion. Their system was able to detect 10 landmarks on the segmented images.

Chen *et al.* [20] used a multiplayer perceptron (MLP) together with genetic algorithms (GA) to improve the performance of the landmarking system. The MLP was trained to search for sub-images that contained each of the landmarks. The GA was used to speed up the process based on the fitness evaluated using MLP. However, they did not specify the accuracy of the locations of the landmarks or the sizes of the sub-images obtained.

Recently, Innes [21] used Pulse Coupled Neural Networks (PCNN) to highlight regions containing key craniofacial features from digital x-rays. They applied different size averaging filters prior to using the PCNN to minimize the noise in the different regions of the image. In this study a bigger set of images (109 x-rays) was used and tested on three points with a success rate of 36.7%, 88.1% and 93.6%, respectively. Although PCNNs have been shown to be capable of image smoothing and segmentation, they require a large amount of manual intervention to set the required parameters.

1.3 Research Objectives

Hand tracing radiographs for cephalometric analysis can be a tedious and time-consuming task requiring extensive training and experience. If the whole system could be automated, orthodontists could concentrate on planning treatment and discussing the proposed changes with their patients. The main objective of this research is to automate the entire detection process. This is achieved by deriving the expectation window or region of interest for each landmark based on the observation that different x-ray images exhibit different characteristics, and no two images are exactly the same. A priori knowledge of the expected location of the landmark based on the knowledge of the orientation, the approximate size

and the relative position of the x-ray is exploited to improve accuracy and robustness of detection. Automated cephalometric evaluation has been subject of research for many years and most of the existing methods suffer from low recognition rates and small test sets, and their performance is highly correlated to the quality of the x-ray image. In this research, two novel methods based on the use of neuro-fuzzy and neural networks are introduced. In the first algorithm, neuro-fuzzy networks are introduced for minimizing the search windows. Template matching techniques are then applied to pin point the exact locations of the landmarks. In the second algorithm, multi layer perceptron is used as a function approximator to directly predict the location of the landmark.

1.4 Thesis Organization

This thesis is organized as follows. Chapter 2 provides an introduction to cephalometry and cephalometric measurements including definition of landmarks, their locations and lines used in cephalometric evaluation. Chapter 3 is a brief introduction to neural networks, different architecture and different training techniques. We also provide a brief introduction to fuzzy logic and fuzzy sets with the advantages and disadvantages of both systems. In Chapter 4 we present fuzzy systems and neuro-fuzzy networks, which are going to be used in the localization system. Chapter 5 is a presentation of the techniques we use for extraction of features and the method used for clustering the data and reduction of the dimension of the feature vector. A detailed description of the method used in localization of landmarks using Multi Layer Perceptron is presented in Chapter 6. In Chapter 7, we present the experimental results and a comparison of results obtained by the two systems is presented. We also provide a comparison of results obtained here with those obtained by other researchers. Chapter 8 contains conclusions, contributions and future work.

Chapter 2

Cephalometrics

2.1 Introduction

Cephalometry is the scientific measurement of the dimensions of the head, usually through the use of standardized lateral skull radiographs, or cephalograms. Analysis of the cephalogram images and measurement of parameters is based on a set of agreed upon feature points called landmarks. The detection of the landmarks plays an essential role in diagnosis and planning of treatment by orthodontists. Lines are constructed from the detected landmarks, and the subsequent measurement of various angular and linear parameters are measured and recorded. This allows comparison with standard values, and aids in diagnosis and assessment of craniofacial growth, planning orthodontic treatment, effects of treatment and evaluation of treated cases.

2.2 History

X-rays are the electromagnetic radiation of a specific wavelength between gamma and ultraviolet waves. An x-ray fires a beam of x-radiation, which travels through

the head and produces an image on medical x-ray film of the part immediately below the beam. The patient is placed in a cephalostat with their head oriented at 90^{o} to the X-Ray beam at a distance of 2-4 meters from the tube. The film is placed 30-40 cm from the head to produce an x-ray picture (Fig. 2.1). This is a standard under which all cephalometric radiographs are taken worldwide. It ensures that radiographs taken at different centers are directly comparable.

The first x-ray pictures of the skull in the standard lateral view were taken in 1922. It was not until 1931 that Hofrath and Broadbent [22] developed standardized method for the production of cephalometric radiographs, which used special holders known as cephalostats to permit assessment of growth and of treatment response. By introducing the x-ray pictures of the skull cephalometric radiology was born. Since then, a variety of analysis based on a standard lateral radiograph of the skull has been developed. Brodie [23] applied a method based on landmarks to quantify malocclusions and study their effect. Landmarks were defined using cephalometric radiographs. In 1948 Downs [24] introduced the first cephalometric analysis method. Downs selected ten angular measurements on the lateral radiographs from selected groups of individuals and gave them a clinical significance. His method has been the basis for many of the methods used at present.

Figure 2.1 A typical cephalogram of the lateral skull x-ray

2.3 Cephalometric Tracing

Cephalograms are rarely used directly for analysis. Instead, a tracing as shown in Fig. 2.2 is first made to outline the most important bony and soft visible tissues. Tracings from previous or subsequent x-rays are superimposed to compare the growth and monitor the treatment.

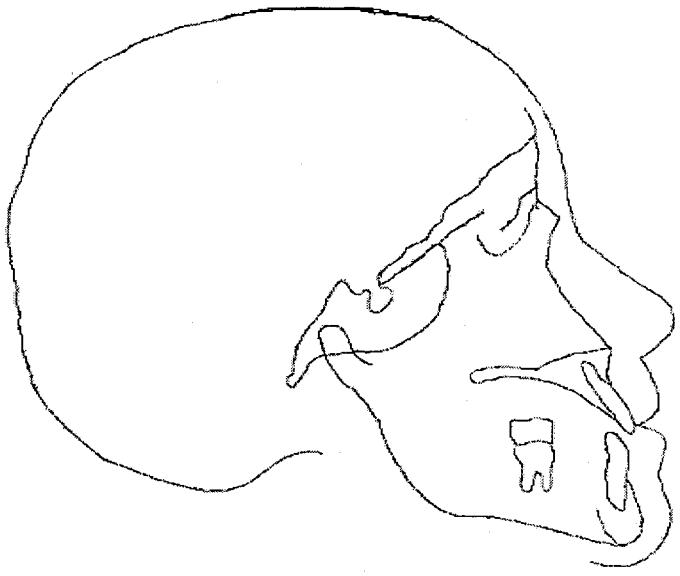

Figure 2.2 A Cephalometric tracing of soft and bony tissue outlines

2.4 Landmarks

The effective evaluation of radiographs depends on accurate definition and localization of landmarks, which provides the basis for all further analysis. The location of landmarks depends on a number of factors such as age, sex, growth and race. The shapes of landmarks may differ in two x-rays of the same patient. A slight shift in the orientation of the skull will generate a shift in the overlapping features. The ease of locating landmarks depends on the quality of the x-ray which is marred by magnification and distortion. Magnification is a result of the

divergence of the x-rays because of the difference in object-image distances. Distortion arises from the representation of three-dimensional objects in two-dimensional form. Another source of difficulty in locating landmarks is overlapping of anatomical features. Accurate centering and positioning of the head will eliminate some of the difficulties but the experience and practice of the observer plays a major role in interpreting radiographs.

Table 2.1 lists some of the most commonly used landmarks according to [1]. The selection of the landmarks depends on personal prefernces of the orthodontist. Fig.2.3 shows the postion of landmarks on the cephalogram and Fig.2.4 shows the landmarks on cephalometric tracing.

Table 2.1 Most commonly used landmarks and their definition.

No	Abbrevation	Defination
1	N	*Nasion.* The most antrior point of the nasofrontal in the median plane. The skin nasion (N') is located at the point of maximum convixity between nose and torehead.
2	S	*Sella.* The midpoint of the hypophysial fossa.
3	Se	*Midpoint of the entrance to sella.* The midpoint of the line conecting the posterior clinoid process and the anterior opening of the sella turicica.
4	Sn	*Subnasal.* A skin point; the point at which the nasal septum merger mesialy with the integument of the upper lip.
5	A	*Point A, subspindale.* The deepest midline point in the curved bony outline from the base to the alveolar process of the maxilla.
6	APMax	The *anterior* landmark for *determining the length of the maxilla.* It is constructed by dropping a perpendicular from A to the palatal plane.
7	Pr	*Prosthion*, The lost most anterior point on the alveolar portion of the premaxilla, in the median plane, between the upper central incisors.
8	Is	*Incisor superius.* Tip of the crown of the mostanterior maxillary central incisor.
9	Ap1	*Apicale.* Root apex of the most anterior maxillary central incisor.
10	Ii	*Infradentale.* Alveolar rim of the mandible; the lightest, most anterior point on the alveolar process, in the median plane, between the mandibular central incisors.

11	Ap1	*Apicale.* Root apex the most anterior mandibular central incisor.
12	Id	*Infradentale.* Alveolar rim of the mandible; the highest, most anterior point on the alveolar process, in the median plane, between the mandibular central incisors.
13	B	*Point B, superamentale.* Most anterior part of the mandibular base. It is the most posterior point in the outer contour of the mandibular alveolar process, in the median plane.
14	Pog	*Pognion.* Most anterior point of the bony chin, in the median plane.
15	Gn	*Gnathion.* The most anterior and inferior point of the bony chin.
16	Go	*Gonion.* A constructed point, the intersection of the lines tangent to the posterior margin of the ascending rams and the manibular base
17	Me	*Manton.* The most caudal poit in the ouline of the symphysis; it is regarded as the lowest point of the mandible and corresponds to the anthropological gnathion.
18	APMan	The *anterior* landmark for *determining the length of the mandible.* It is defined as the perpendicular dropped from Pog to mandibular plane.
19	ar	*Articular.* The point of intersection of the posterior margin of the ascending and the outer margin of cranial base.
20	Cd	*Condylion.* Most superior point on the head of the condyle.
21	Or	*Orbitale.* Lowermost point of the orbit in the radiograph.
22	Pn/2	A *Constructed point.* It is obtained by bisecting the Pn vertical, between its intersection with the palatal plane and point N'.
23	Int.FH/R.asc	*Intersection* of the ideal Frankfurt horizontal and posterior margin of ascending armus.
24	ANS	*Anterior nasal spine.* The tip of the bony anterior nasal spine, in the median plane.
25	PNS	*Posterior nasal spine.* The intersection of a continuation of the anterior wall of the pterygopalatine fosa and the floor of the nose
26	S'	*Landmark* for assessing the length of the maxillary base, in the posterior section. It is defined as a perpendicular dropped from point S to a line extending the palatal plane.
27	APOcc	*Anterior point for the occlusal plane.* The midpoint in the incisor overbite in occlision.
28	PPOcc	*Posterior Poin of the occlusal plane.* The most distal point of contact between the most posterior molar in occlusion.
29	Ba	*Basion.* Lowest point on the anterior margin of the foramen

		magnum in the median plane.
30	Ptm	*Pterygomaxillary fissure.* The contour of the fissure projected onto the palatal plane.

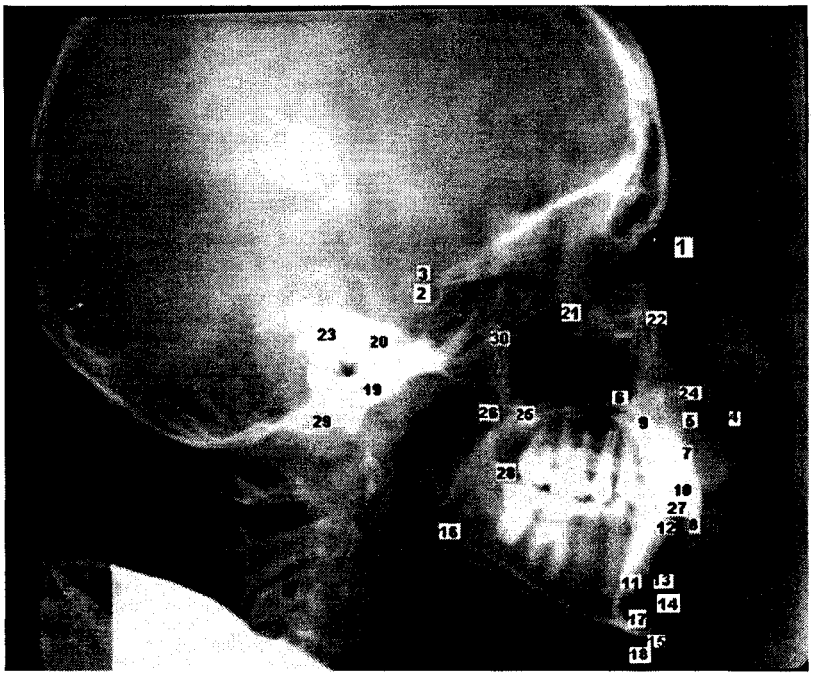

Figure 2.3 Most commonly used landmarks as defined on cephalograms

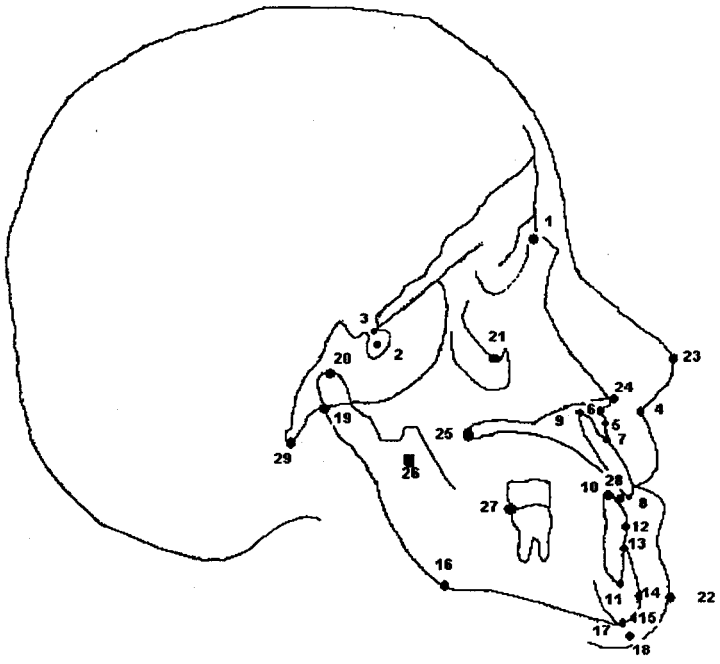

Figure 2.4 Most commonly used landmarks as defined on the cephalometric tracing.

2.5 Reference Lines

The above-mentioned landmarks are used to construct a considerable number of lines. Fig 2.5 shows the most frequently used lines and Table 2.2 lists the lines and their definitions.

Table 2.2 The most commonly used reference lines

No.	Line	Definition
1	S-N Sella-nasion	Anteroposterior extent of anterior cranial base (Se-N)
2	S-Ar	Lateral extent of cranial base
3	ar-Go	Length of ramus (1st measurement)
4	Me-Go	Extent of mandibular base (1st measurement)
5	N-A	Nasion–point A
6	N-B	Nasion–point B
7	N-Pr	Nasion- prosthion
8	N-Id	Nasion- infradentale
9	N-Pog	Nasion - pogonion
10	N-Go	Nasion- gonion line, for analysis of the gonial angle
11	Pal	Pal Palatal plane (ANS-PNS)
12	Occ	Occlusal plane (APOcc-PPOcc)
13	S-Gn	Y-axis
14	S-Go	Posterior facial height
15	I-SN	Long axis of upper incisor to SN
16	I- Pal	Long axis of upper incisor to Pal
17	I-MP	Long axis of lower incisor to mandibular plane
18	ManBase	Extent of mandibular base (Go-Gn, 2nd measurement)
19	MaxBase	Extent of maxillary base (APMax-PNS)
20	R.asc	Cd-Go length of ramus (2nd measurement)
21	S-S'	Perpendicular from point S (starting from the SN line) to point S'
22	Pn line	Perpendicular to SeN, drawn from the soft tissue nasion (N) as Pal.
23	'H' line	Modified Frankfurt horizontal; parallel to the SeN line which bisects the Pn line from N to Pal (Pn/2-fH/R.asc.)
24	EL	Aesthetic line. Tip of nose -soft tissue pogonion

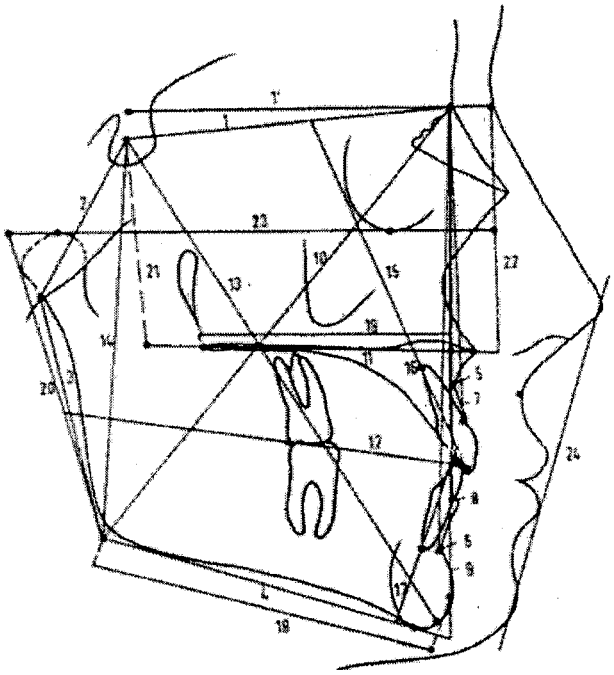

Figure 2.5 The most frequently used lines in cephalometry

2.6 Angular Measurements

The reference lines enable making the angular and linear measurements and determine the dimensions of the radiograph. Table 2.3 lists the most frequently used angles and their mean values. Fig. 2.6 shows the most frequently determined angles in the cephalometric tracings.

Table 2.3 The most frequently used angles and their mean values

No.	Points of the angle	Definition	Mean Value
1	N-S-Ar	Saddle angle	123°± 5°
2	S-Ar-Go	Articular angle	143°± 6°
3	Ar-Go-Me	Gonial angle	128°± 7°
4	Sum	Sum of sella, articular and gonial angles	394°
5	Ar-Go-N	GO1, upper gonial angle	52°-55°
6	N-Go-Me	G02, lower gonial angle	70-75°
7	SNA 810	Anteroposterior position of maxilla	81°
8	SNB	Anteroposterior position of mandible	79°
9	ANB	Difference between SNA and SNB	2°
10	S-N-Pr	Anteroposterior position of alveolar part of premaxilla	84°
11	S-N-Id	Anteroposterior position of alveolar part of mandible	81°
12	Pal-MP	Angle between palatal and mandibular plane	25°
13	Pal-Occ	Upper occlusal plane angle	110°
14	MP-Occ	Lower occlusal plane angle	14°
15	SN-MP	Angle between SN and mandibular plane	32°
16	Pn-Pal	(∠ of incl.) Angle of inclincation after A.M. Schwarz	85°
17	N-S-On	(Y-axis) Angle between SN line and SGn line, anteriorly	66°
18	$\underline{1}$-SN	Angle between upper incisor axis and SN line posteriorly	102°
19	$\underline{1}$-Pal	Angle between upper incisor axis and palatal plane, anteriorly	70°±5°
20	$\bar{1}$-MP	Angle between lower incisor axis and mandibular plane, posteriorly	90°±3°
21	ii angle	Interincisal angle between upper and lower central incisor axes, posteriorly	135°

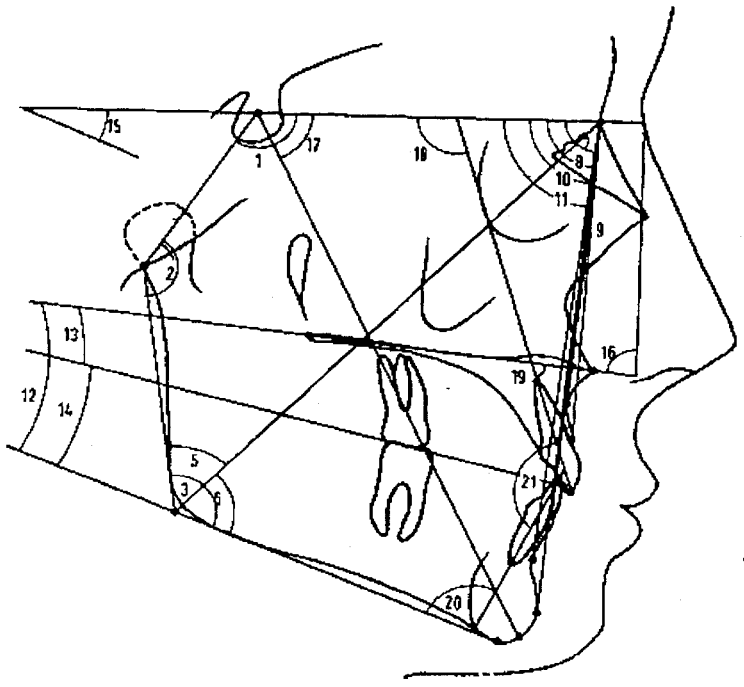

Figure 2.6 The most frequently determined angles

2.7 Linear Measurements

There are two types of linear measurement.

1. Point to Point: Distance between two points.
2. Point to Line: The perpendicular distance from a point to a line.

Table 2.4 shows the commonly used linear measurements and Fig. 2.7 shows the linear measurements on a cephalometric tracing.

Table 2.4 The most commonly used reference lines

No.	Distance	Definition	Mean Value
1	S-N (SeN)	Anteroposterior extent of anterior cranial base	71mm
2	S-Ar	Extent of lateral cranial base	32-35 mm
3	S-Go	Posterior facial height	
4	N-Me	Anterior facial height	
5	MaxBase	Extent of maxillary base, correlated with Se-N	
6	ManBase	Extent of mandibular base, correlated with SeN	
7	R.asc.	Extent of ascending ramus, correlated with SeN	
8	S'-F. Ptp.	Distance from S' to projection of the anterior wall of the pterygopalatinal fossa onto the palatal plane, expression for anteroposterior displacement of the maxillary base	
9	S-S'	Expression for deflections of the maxillary base	42-57mm
10	$\underline{1}$ -N-Pog	Distance from incisal edge of. $\underline{1}$ to N-Pog line	
11	$\overline{1}$ -N-Pog	Distance from incisal edge of $\overline{1}$ to N-Pog line	

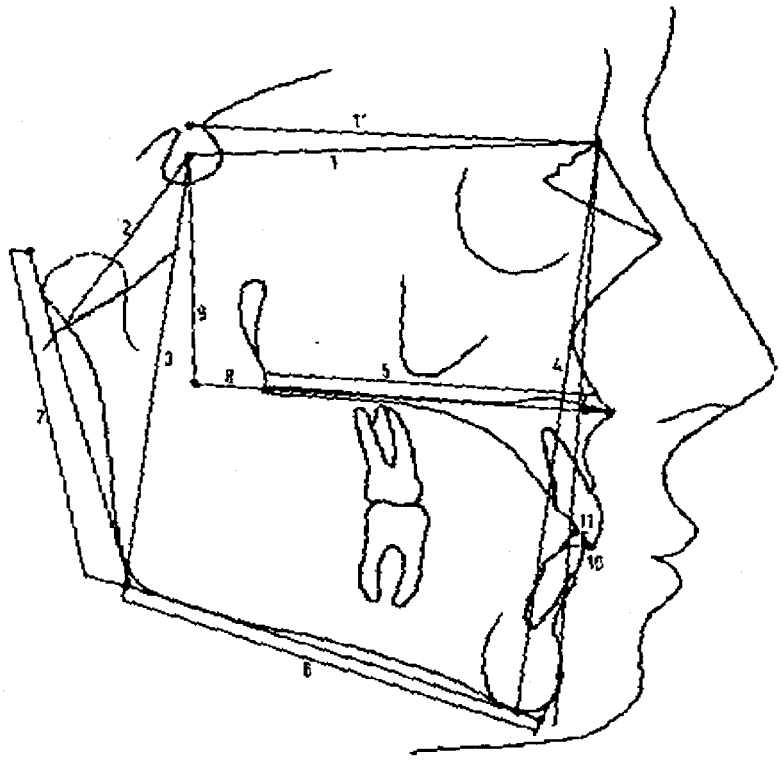

Figure 2.7 The principal linear measurements used in cephalometric analysis

Chapter 3

Neural Networks and Fuzzy Systems

3.1 Neural Networks

3.1.1 Introduction

An artificial neural network (ANN) is an information-processing structure that tries to imitate the human being's abilities in perception, vision, associative memory, and pattern recognition. ANN is an interconnected assembly of simple processing elements (PE), called *units* or *nodes*, whose function is based on that of neurons in the human brain. Mathematically, Each neuron is represented by a model that emulates some of the observed properties of biological nervous systems, and draw a conclusion based on the analogies of adaptive learning in human brain. The processing ability of the network is stored in the connection strengths between neurons, or *weights*. The weights are obtained by a process of adaptation to, or *learning* from, a set of labeled data or training patterns. The key element of the ANN paradigm, is the novel structure of the information processing system. ANNs are usually composed of a large number of highly interconnected PE's that are analogous to neurons and are tied together with weighted connections

that are analogous to synapses. The learning in ANN's is similar to the learning in humans. In ANN's, the learning typically occurs by examples, through training using a set of inputs and outputs.
In ANN's training algorithms iteratively adjust the connection weights between nodes, while learning in humans systems involves adjustments to the synaptic connections that exist between neurons.

The knowledge necessary to solve specific problems is stored in the connection weights. ANN's have faster responses and higher performance than those of the sequential digital computers in emulating the capabilities of the human brain.

Although ANN's have been around since the late 1950s, it wasn't until the mid-1980s that algorithms became advanced enough to solve applications of general nature. Today ANN's are being applied to an increasing number of real-world problems of considerable complexity. They are good pattern recognition engines and classifiers, with the ability to generalize in making decisions about imprecise input data. ANN's offer ideal solutions to a variety of classification problems such as speech, character and signal recognition, as well as functional prediction, function approximators and system modeling where the actual process is not completely understood or is highly complex. ANN's may also be applied to control problems, where the input variable's measurements are used to drive an output actuator, and the network learns the control function. The advantage of ANN's lies in their ability to perform well in the presence of noise or incomplete input data and their ability to learn. They are often good at solving problems that are too complex for conventional technologies (e.g., problems that do not have an analytical solution or those which for the solution is too complex) and are often well suited to problems that humans are good at solving, but for which traditional methods are not.

3.1.2 Architecture of Artificial Neural Networks

Generally speaking, ANNs can be classified into two categories: feedforward and feedback structures. Feedforward ANNs are layered networks for which the flow

of information is in one direction from input to output. In feedback ANNs information can flow in both directions.

3.1.2.1 Feedforward Neural Networks

Feedforward ANN's (Fig 3.1) allow signals to travel in one way only from input to output. There is no feedback (loops) i.e. the output of any layer does not affect neurons of the same or previous layer. Neurons in each layer can accept information only from neurons in the previous layer. Feed-forward ANN's tend to be straightforward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down. The processing in this type of network is clear and easy to implement. However, the number of neurons and the number of training samples required are very high.

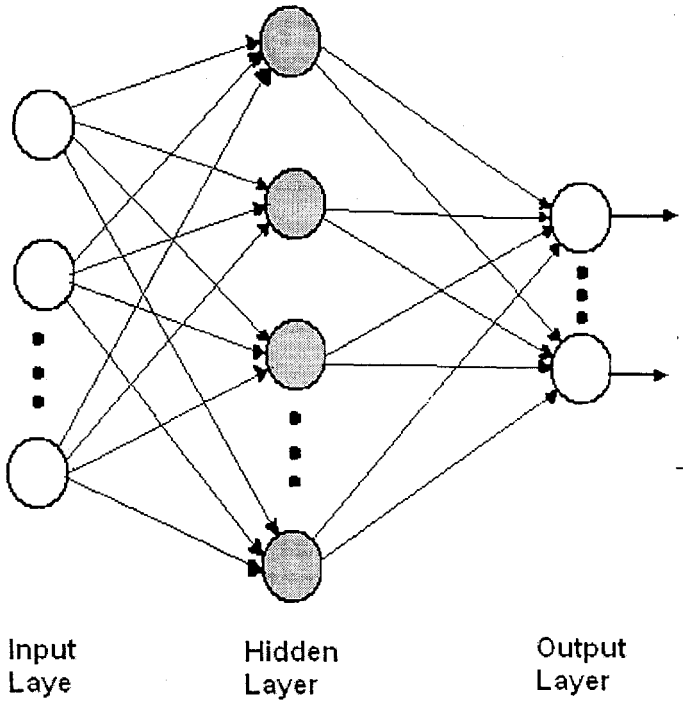

Figure 3.1 An example of a simple feedforward network

3.1.2.2 Feedback Neural Networks

Feedback networks (Fig 3.2) can have signals traveling in both directions by introducing loops in the network. The signal is transferred between neurons many times while the state of the network keeps changing. Feedback networks are dynamic; their 'state' is changing continuously until they reach a state of equilibrium. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as recursive or recurrent. Feedback networks are very powerful and can get extremely complicated, but the number of neurons and training samples are fewer than that required for the feedforward for solving the same problem. Similar to all recursive systems feedback networks suffer from the problem of stability.

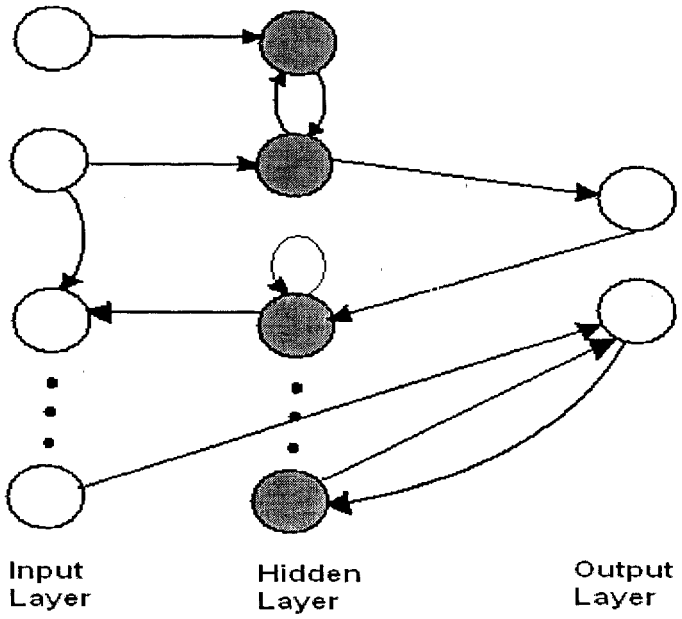

Figure 3.2 An example of a feedback network

3.1.3 Artificial Neuron

Neural networks use a set of processing elements (or nodes) to represent the function of neurons in the brain. (Hence, the name, neural networks.) These neurons are interconnected in a network that can then learn to identify patterns in

data as it is exposed to the data in training sets. The neurons process information as follows: the signal appears at the input units. The effect that the signal makes may be approximated by multiplying the signal by some number or *weight* to indicate the strength of the synapse. Next the weighted signals are summed to produce an overall unit *activation*. If this activation exceeds a certain threshold the unit produces an output response Fig. 3.3. This functionality is captured in the artificial neuron known as the Threshold Logic Unit (TLU) shown in Fig 3.3. This neuron was originally proposed by McCulloch and Pitts [25].

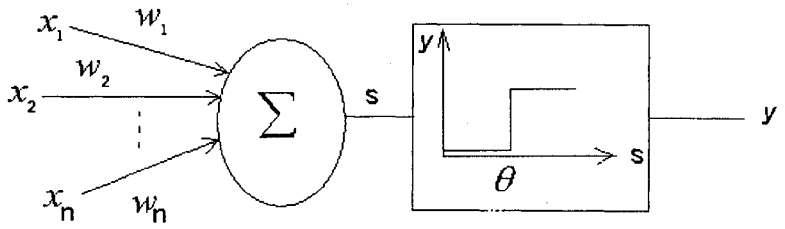

Figure 3.3 A Threshold Logic Unit (TLU)

If there are *n* inputs with signals $X = [x_1, x_2, ..., x_n]^T$ and weights $w_1, w_2, ..., w_n$. then

$$S = x_1 w_1 + x_2 w_2 + ... + x_n w_n = \sum_{i=1}^{n} x_i w_i \tag{3.1}$$

$$y = f(s) = f(\sum_{i=1}^{n} w_i x_i + \theta) \tag{3.2}$$

where θ is the threshold.

The output *y* is then given by thresholding the activation

$$y = \begin{cases} 1 \; if \; s \geq \theta \\ 0 \; if \; s < \theta \end{cases} \tag{3.3}$$

which is commonly called a *signum function*

The structure shown in Fig. 3.3 also known as the preceptron is basically a linear classifier which is capable of separating two disjoint classes as shown in Fig 3.4. The process of determining the weights is called the learning scheme [26].

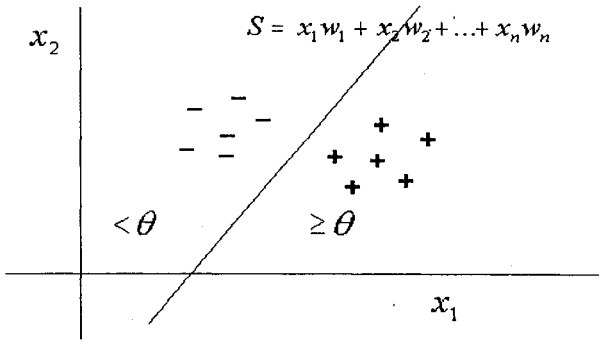

Figure 3.4 A classifier for two classes

The preceptron has limitation; it can only solve only linearly separable problems. Non-linearly separable problems cannot be solved using the preceptron. For non-linearly separable problems, Multi Layer Perceptron (MLP) or Radial Basis Function (RBF) is used.

3.1.4 Learning Methods

ANN's learn patterns by adjusting the connection weights between neurons. All learning methods used for adaptive neural networks can be classified into two major categories: Supervised Learning and Unsupervised Learning. If the learning phase is distinct from the operation phase, it is said that the neural network learns off-line. If the learning phase and the operation phase are performed at the same time, then the network is learning on-line. Usually, supervised learning is performed off-line, whereas unsupervised learning is performed on-line. In both learning methods the training is performed by

adjusting a set of weights connecting the neurons in the different layers of the ANN.

3.1.4.1 Supervised Learning

This kind of learning incorporates an external teacher, so that each output unit is told what its desired response to input signals should be. For supervised learning a set of input data and its expected response is needed. There are several paradigms for supervised learning such as error-correction learning, reinforcement learning and stochastic learning. An important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of the cost function, which is a measure of the difference between the actual response of the network and the expected or the desired response. The aim is to determine a set of weights which minimizes the cost function. One well-known method, which is common to many learning paradigms, is the least mean square (LMS). Supervised training is analogous to a student guided by an instructor

3.1.4.2 Unsupervised Learning

In unsupervised learning, no external teacher is used. The learning is based upon only local information. It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their properties. This is analogous to a student who tries to learn the lesson totally on his or her own. Hebbian learning and competitive learning methods are paradigms of unsupervised learning. Unsupervised algorithms essentially perform clustering of the data into similar groups based on the input features or attributes [27].

3.1.5 Advantages of Neural Networks

Neural networks and artificial intelligence based models have potential advantages over other models due to following reasons:

(1) Once trained, ANNs have powerful computing abilities and very high processing speed via massive parallel structures.

(2) ANNs have fault tolerance because they are constructed from a large amount of neurons. They are not sensitive to missing, noisy or corrupted data.

(3) ANNs start processing the data without any previous assumption. They start with random weight assignment to various connection weights. Adjustments are made based on the difference between predicted and desired output. This is basically finding the relationship between the input and output, which make the ANN's good function approximators.

(4) Some ANN models can be retrained using additional inputs. Once trained they can be used to predict an output of an input never seen before.

(5) There are several neural network models available to choose from for use in a particular problem.

(6) ANNs can solve complicated problems that have unclear reasoning rules.

3.1.6 Disadvantages of Neural Networks

(1) Training is quite time consuming and convergence of the error to the required accuracy is not always guaranteed.

(2) No set of rules are available for network selection.

(3) The knowledge in ANN's is implicitly represented, making it hard to understand

The implicit representation of knowledge in ANN's is the most difficult and critical task in its analysis and design due to the lack of an appropriate language for knowledge representation. In this research, ANN's and neuro-fuzzy networks are used in solving the problem of localization of craniofacial landmarks. MLP is used as a function approximator. After being trained using a

set of labeled images it is then used in locating landmarks in images outside the training set.

Fuzzy set theory is introduced to neural network due to its advantages in knowledge representation. The knowledge represented in the connection weights is described by a set of fuzzy inference rules. In the next section, the basic ideas and features of the fuzzy systems will be described.

3.2 Fuzzy Logic

Many decision-making and problem-solving tasks are too complex to be understood quantitatively; however, people succeed by using knowledge that is imprecise rather than precise. Fuzzy set theory resembles human reasoning in its use to approximate information and uncertainty to generate decisions. It was specifically designed to mathematically represent uncertainty and vagueness and provide formal tools for dealing with the imprecision in many problems. Since knowledge can be expressed in a more natural way by using fuzzy sets, many engineering and decision problems can be greatly simplified by using fuzzy logic. Fuzzy logic is a superset of the conventional (Boolean) logic that has been extended to handle the concept of partial truth -- truth-values between "completely true" and "completely false".

Fuzzy logic was known during the time of Plato; however, it did not get much attention until 1965 when Lotfi A. Zadeh published his work in "Fuzzy Sets" [28-30], in which he described the mathematics of fuzzy set theory, and its extension of fuzzy logic. This theory proposed making the membership function (or the values False and True) to operate over a range of real numbers [0.0, 1.0]. New operations for the calculus of fuzzy logic were proposed, and showed to be a generalization of classic logic. It is this theory, which we will now discuss.

3.2.1 Fuzzy Sets Theory

Fuzzy set theory is an implementation of classes or groupings of data with boundaries that are not sharply defined (i.e., fuzzy). In classical set theory or crisp theory, data sets are represented by "crisp" definitions. Any member can either "belong" or "do not belong" to a dataset. In a fuzzy set any member can belong to the data set with a certain value or membership. The process of finding this membership called "fuzzification" which is generalizing the concept of a crisp set to a fuzzy set with blurred boundaries. The benefit of extending crisp theory and analysis methods to fuzzy techniques is the strength in solving real-world problems. It is inevitable that these real-world problems have some degree of imprecision and noise in the variables and parameters measured and processed for the application. Accordingly, linguistic variables are a critical aspect of some fuzzy logic applications, where general terms such a "large," "medium," and "small" are each used to capture a range of numerical values. While similar to the conventional quantization, fuzzy logic allows these stratified sets to overlap (e.g., an 85 kilogram man may be classified in both the "large" and "medium" categories, with varying degrees of belonging or membership to each group). Fuzzy set theory encompasses fuzzy logic, fuzzy arithmetic, fuzzy mathematical programming, fuzzy topology, fuzzy graph theory, fuzzy data analysis and fuzzy neural networks, though the term fuzzy logic is often used to describe all of these.

The definition of a fuzzy set is can be given as follows:

A fuzzy set A, is a subset of the universe of discourse, X, that admits partial memberships. The fuzzy set A is defined as an ordered pair $A = \{x, \mu(x)\}$, where $x \in X$, and $0 \le \mu_A(x) \le 1$. The membership function $\mu_A(x)$ describes the degree to which the object x belongs to the set A, $\mu_A(x) = 0$ represents no membership, and $\mu_A(x) = 1$ represents full membership

As an example, let X represent the age of all people. The subset A of X represents people who are young. The subset A is a fuzzy subset with a membership function shown in Fig. 3.5. [31]. As can be seen from Fig. 3.5 a person who is 25 years old has a membership of 1.0 while another person who is

50 years old has a membership of 0.5. Using crisp logic a person is 25 years old or younger is considered young and not young otherwise. This will mean that someone who is 24.99 is considered young while someone who is 25.01 is consider old.

another example if we consider a fuzzy set A representing all numbers that are *close to 6,* since "close to 6" is a fuzzy statement and there is no unique membership function representing the set A. The designer has to decide what the membership should be. In crisp logic the membership can be either 1 or 0 while in fuzzy logic every set has several ways of representing it. Fig. 3.6 shows two ways of representing the membership function that represents the fuzzy statement "close to 6" [Bedeck][32]

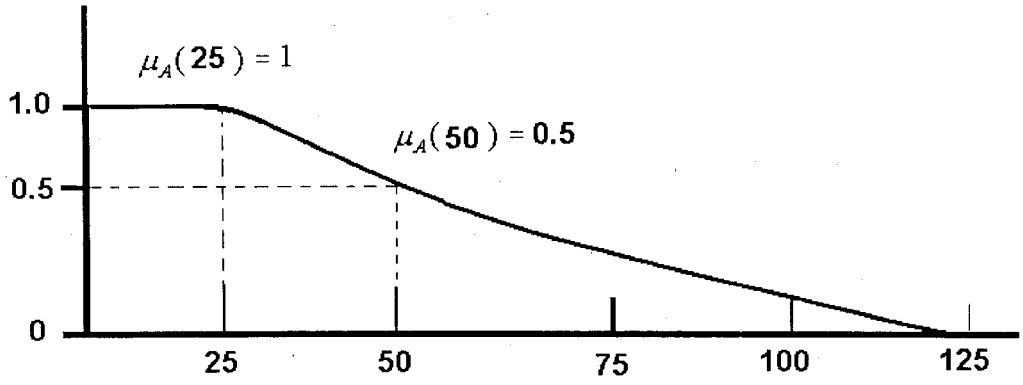

Figure 3.5. The membership function describing the relation between the person's age and the degree to which a person is considered young.

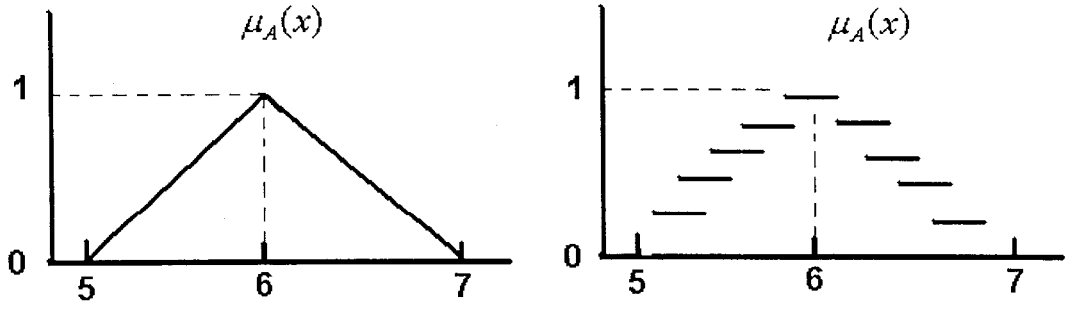

Figure 3.6. Two representation of membership function of the fuzzy set A that represents " real numbers close to 6"

3.2.2 Properties of Fuzzy Sets

The operations on fuzzy sets are an extension of those of traditional sets. The common operations of fuzzy sets include *union, intersection, complement, comparison, and containment,* [33].

Let X be the universe of discourse, $A \in X$ and $B \in X$ the following operations are defined.

1. ***Union*** **(A $\cup$ B):** $\mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x)) \ \forall x \in X$ (3.4)

2. ***Intersection*** **(A $\cap$ B):** $\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x)) \ \forall x \in X$ (3.5)

3. ***Complement*** **($\overline{A}$):** $\mu_{\overline{A}}(x) = 1 - \mu_A(x) \ \forall x \in X$ (3.6)

4. ***Equivalence Relation:*** $A = B \ iff \ \mu_A(x) = \mu_B(x) \ \forall x \in X$ (3.7)

5. Containment: $A \subset B iff \ \mu_A(x) \prec \mu_b(x) \ \forall x \in X$ (3.8)

In addition to these operations, De Morgan's law, the distributive laws, algebraic operations such as addition and multiplications, and notation of convexity have fuzzy set equivalents [34].

Table 3.1 Properties of fuzzy sets

Commutativity	$A \cup B = B \cup A, A \cap B = B \cap A$
Distributivity	$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$
De Morgan's Laws	$\overline{(A \cup B)} = \overline{A} \cap \overline{B}, \overline{(A \cap B)} = \overline{A} \cup \overline{B}$
Algebraic Sum $A + B$	$\mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x).\mu_B(x)$ $\forall x \in X$
Algebraic Product $A \otimes B$	$\mu_{A\otimes B}(x) = \mu_A(x).\mu_B(x) \ \forall x \in X$
Bounded Sum $A \oplus B$	$\mu_{A\oplus B} = \min(1, \mu_A(x) + \mu_B(x)) \ \forall x \in X$
Bounded Difference $A \ominus B$	$\mu_{A\ominus B} = \max(0, \mu_A(x) - \mu_B(x)) \ \forall x \in X$

3.2.3 Membership Functions

A fuzzy set is completely characterized by its membership function. Since most of the fuzzy sets in use have a universe of discourse X it is impossible to list all the pairs defining the membership function. A more convenient way is to express the membership function in a mathematical form. In this section we list the most popular mathematical forms for representing membership functions.

3.2.3.1 Triangular Membership Functions

A triangular membership function is defined as follows:

$$\mu(x) = \begin{cases} 0 & x \le a \\ \dfrac{x-a}{\theta-a} & a \le x \le b \\ \dfrac{b-x}{b-\theta} & b \le x \le \theta \\ 0 & if\ x > b \end{cases} \tag{3.9}$$

where the parameters a and b represent the left and right boundary of the set and θ represent the center of the set. Fig. 3.7 shows the triangular membership

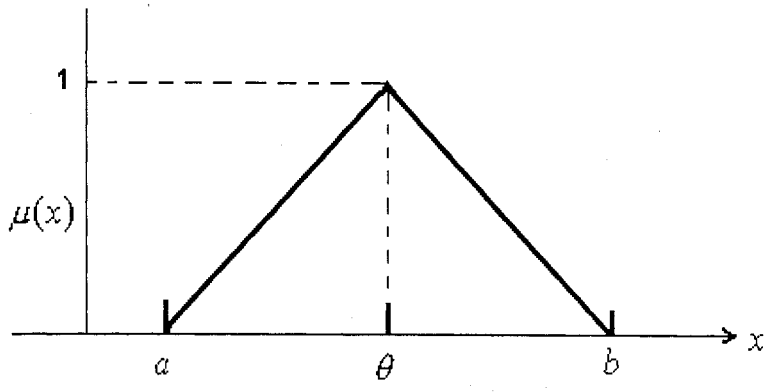

Figure 3.7 A Triangular membership function

3.2.3.2 Gaussian Membership Functions

The Gaussian membership function is defined by

$$\mu(x) = \exp(-\frac{1}{2}(\frac{x-c}{\sigma})^2), \tag{3.10}$$

where c is the center of the membership function and σ is the width as shown in Fig. 3.8

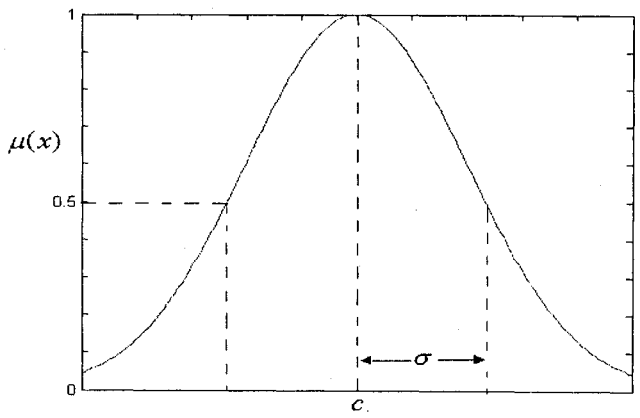

Figure 3.8 A Gaussian membership function

3.2.3.3 Generalized Bell Membership Function

$$\mu(x)\frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}}, \tag{3.11}$$

where *c* is the center of the membership function, *a* is the width and *b* is the slope of the curve as shown in Fig. 3.9

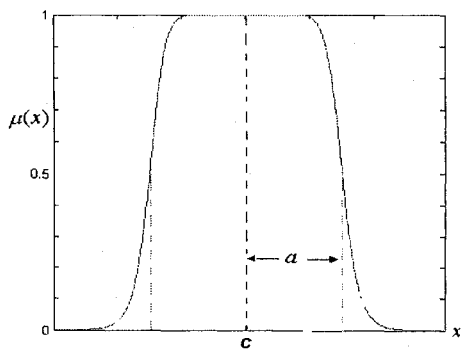

Figure 3.9 A generalized bell membership function

3.3 Fuzzy Systems

A fuzzy system is a system that uses fuzzy logic and fuzzy set theory instead of Boolean logic. In other words, a fuzzy system is a collection of membership functions and rules that are used to reason about data. Among all fuzzy systems, fuzzy rule-based systems are the most successfully used in many problems such as decision-making systems, expert systems and control systems. Fig. 3.10 shows a typical example of fuzzy rule-based system. A fuzzy rule-based system consists of 4 main parts: Fuzzifier, Knowledge Base, Inference Engine and Defuzzifier. The first part is the fuzzifier, which converts input data from crisp values to fuzzy values based on membership of inputs to fuzzy sets. The knowledge base is implemented by fuzzy sets and fuzzy inference rules in the form of "**If x is A then Y is B**". The inference engine implements a fuzzy inference algorithm that deduces results from inference rules and the present inputs. The defuzzifier converts fuzzy data to crisp outputs.

Fuzzy systems store "rules" and estimate sampled functions from linguistics inputs to generate linguistic outputs. Fuzzy methods are powerful in modeling human thinking and perception. They are successful in expressing the human experience in a form that a machine can use, to imitate human pattern

recognition and judgment capabilities and to convert the information into a form that people can understand.

3.3.1 Advantages of Fuzzy Systems

(1) Fuzzy systems can mimic human decision making to handle vague concepts.

(2) Rapid computation due to intrinsic parallel processing nature.

(3) Improved knowledge representation and uncertainty reasoning with natural language processing/programming capabilities to model complex, non-linear problems.

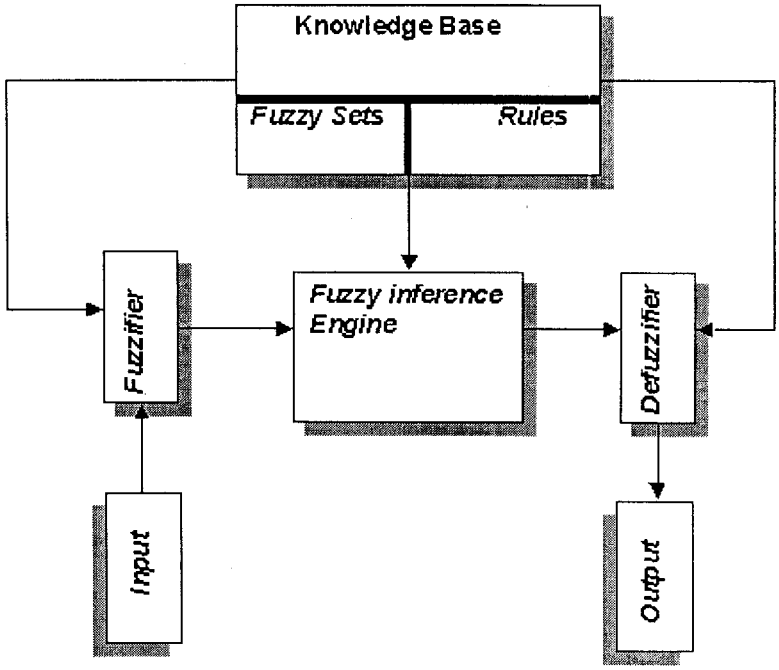

Figure 3.10 Typical fuzzy rule-based systems

3.3.2 Limitations of Fuzzy Systems

(1) Lack of self-organizing, self-tuning and learning mechanisms.

(2) Selection of number of rules and membership functions is done on trial and error bases.

3.4 Conclusion

ANNs are very useful in pattern recognition, since they have the ability to learn; however, their implicit representation of the knowledge makes them very hard to understand. In fuzzy systems the knowledge is represented by fuzzy rules. If a system can have the learning ability of ANNs and the knowledge representation of the fuzzy system, it will combine the advantages of both systems and overcome the first limitation. If the system can decide on the rules and self tune its membership function, then it will overcome the second limitation. Neuro-Fuzzy systems are such systems that integrate the learning ability of ANNs into fuzzy systems.

Chapter 4

Neuro-Fuzzy Networks

4.1 Introduction

The theories of neural networks and fuzzy sets provide two complementary ways of modeling the human brain. The neural network provides means for modeling the lower level, that is, the physiology of the brain, while fuzzy theory provides a means of modeling the higher level or the psychology of the human brain [34].

While fuzzy logic provides an inference mechanism under cognitive uncertainty, neural networks offer other advantages such as learning, adoption, fault tolerance parallelism and generalization. The marriage of neural networks and fuzzy systems combines the learning abilities of the neural-network with the rule structure of fuzzy systems. The resultant structure is fuzzy-neural or neuro-fuzzy system. Neuro-fuzzy systems are systems that have the properties of neural networks such as fault tolerance, and learning abilities and the properties of fuzzy systems such as high-level fuzzy reasoning and rule thinking. In this chapter we introduce the definition of fuzzy neurons.

neuro-fuzzy systems, fuzzy rules formulation, training algorithms, and propose a neuro-fuzzy structure to be used for estimating search windows in cephalograms.

4.2 Fuzzy Neurons (FN)

Neurons are the information processing units in the human brain. In most of the present ANN's, neurons are simplified threshold units; each can have one of two states, 0 or 1. The two-state neuron with linear weights is too simple to express the interaction of neurons in the human brain.

In order to imitate the function of neurons in the human brain, the models of neurons have adopted important results from biological neural science making fuzzy neurons more complicated than the conventional neurons.

An FN can have *n* inputs $\{x_1,...,x_n\}$ and *m* outputs $\{y_1,...,y_m\}$ as shown in Fig.4.1.

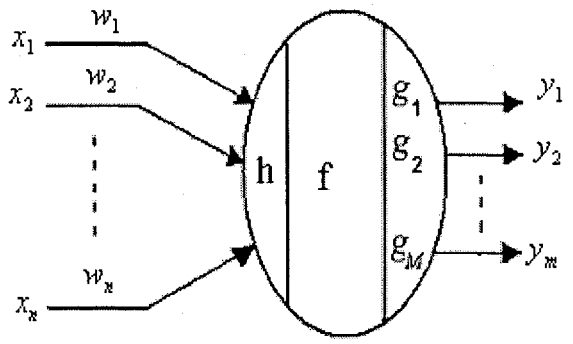

Figure 4.1 A fuzzy neuron model

The input-output relation of a fuzzy neuron can be described as

$$a = h(w_1, x_1, ..., w_n, x_n) \tag{4.1}$$

$$y_i = g_i(f(a,t)) \tag{4.2}$$

where *h()* is the aggregation function, *f()* is the activation function, t is the threshold, w_i { *i*=1,...*n*} are the input weights and { g_j () | j=1,...,*m*} represents the *j*th output function of the neuron. Generally FN's are aggregation operators of T-norms and S-norms instead of the weighted sum in the two-state neuron. The aggregation function and the activation function are the intrinsic properties of the FN, whereas the input weights, output functions, and threshold functions are the learning parameters. Most of the FN's do not use a threshold; the threshold may be contained within the choice of the activation function. The activation function is a mapping operator that transforms the crisp input to a membership value [0,1]. This mapping operation corresponds to a linguistic modifier such as ***VERY*** and ***MORE-OR-LESS.*** The role of this modification is to enhance or diminish the degree to which the inputs give rise to the fuzzy value represented by the fuzzy neuron, before becoming an output [35].

The aggregation function and the activation function determine the characteristics of the fuzzy neuron. Using different aggregation and activation functions result in a fuzzy neuron with different properties. Thus, many different types of fuzzy neurons can be defined by changing the aggregation function. We can define an AND FN if the aggregation is changed to a min operation and OR FN if the min. operation is used as an aggregation function [36].

4.3 Fuzzy Rules

Human beings make decisions based on rules. Although it might not be clear, but all decisions we make are all based on if-then rule. If the weather is nice, then it is a good time to go out. If the weather will be rainy today, but nice tomorrow, then we make a decision not to go out today, and postpone it till tomorrow. Fuzzy systems, which always tend to mimic the behavior of the human brain, work by associating ideas and relating one event to another. However, the decision and the means of choosing that decision are replaced by fuzzy sets and the rules are replaced by fuzzy rules. Fuzzy rules also operate using a series of

if-then statements. For instance, if X then A, if Y then B, where A and B are all subsets of X and Y.

The rules in a fuzzy system with n fuzzy inputs and m outputs are usually of a form similar to the following:

$$R^i : \text{IF } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2}, ..., x_n \text{ is } A_{in} \text{ THEN } y_1 \text{ is } B_1, ..., \text{and } y_m \text{ is } B_m \quad , \qquad (4.3)$$

where $\{x_1, ..., x_n\}$ are input variables (names for know data values), $\{y_1, ..., y_m\}$ are output variables (names for a data values to be computed), and $A_{ij}(j = 1, ..., n)$ and $B_k(k = 1, ..., m)$ are fuzzy subsets.

The part of the rule between the "if" and "then" is the rule's premise or antecedent. This is a fuzzy logic expression that describes to what degree the rule is applicable. The part of the rule following the "then" is the rule's conclusion or the consequece. This part of the rule assigns a membership function to each of one or more output variables. Most tools for working with fuzzy systems allow more than one conclusion per rule. The entire group of rules is collectively known as a rule base or knowledge base. A typical fuzzy system has more than one rule.

4.4 Fuzzy Reasoning

The fuzzification is a mapping of the observed input x_i to the fuzzy sets defined in the corresponding universe $A_{ik}(k = 1, ..., n)$ *and* $B_{ij}(j = 1, ..., m)$. The inference engine is a decision making logic algorithm, which determines the outputs corresponding to the fuzzified inputs with respect to the fuzzy inference rules. The defuzzification produces crisp outputs if needed. Usually, three methods are used: Center of Area, Max Criterion, or Mean of Maximum [37-39]. In the Center of Area method, the crisp value of the output variable is computed by finding the variable value of the center of gravity of the membership function for the fuzzy value:

$$y^{\circ} = \frac{\sum_{i=1}^{n} \mu_i(y) d_i}{\sum_{i=1}^{n} \mu_i(y)} \tag{4.4}$$

where y^o is the crisp output, y_i is the value from the set that has a membership value μ_i [40]. In the Max Criterion method, one of the variable values at which the fuzzy subset has its maximum truth-value is chosen as the crisp value for the output variable. In the Mean of Maximum method, the average of the variable values at which the maximum truth-value occurs is taken as the crisp output.

Example

We will examine the use of fuzzy rules and fuzzy reasoning by an example of controlling an automobile. In this example, the inputs are speed of the automobile and its distance to the automobile in front. The output is breaking strength.
Step1: We first express the experience in form of rules

Rule 1: If the distance between the two cars is SHORT and the car speed is HIGH, then brake HARD for substantial speed reduction.

Rule 2: If the distance between the two cars is LONG and the car speed is HIGH, break MODERATELY HARD

Rule 3: If the distance between two cars is LONG and the car speed is LOW, break SOFT for slower reduction of speed

.
.
Although production rules can be expressed with everyday language, codes are used to simplify the input to the actual Fuzzy Controllers. Replacing linguistic rules with codes is done as follows:

Distance (d) : Long Distance (LD), Medium Distance (MD) and Short Distance (SD).

Speed (s): High Sped (HS), Medium Speed (MS) and Fast Speed (FS).

Breaks Pressure (b): Soft Pressure (SP), Moderate Soft Pressure (MSP), Medium Pressure (MP), Moderate Hard Pressure (MHP) and Hard Pressure (HP).

Rule can be written in a more compact format as in Table 4.1

If s is HS and d is SD then b is HB.

	LS	MS	HS
SD	MB	MHB	HB
MD	MSB	MB	MHB
LD	SB	MSB	MB

Table 4.1 Summary of the fuzzy rules

Step2: Determine membership functions for the antecedent and consequent.

The distance between the two cars and the car speed are the antecedent parts and the braking strength is the consequent part. All parts of the rules are represented by "Fuzzy Sets" expressed through linguistic rules. The distance between the two cars and their speed have a multiple number of fuzzy values and are therefore called "Fuzzy Variables". We define the universe of discourse assuming the speed of both cars is between 0 and 120 km/h. Using triangular membership function, the concept of high speed is represented by Fig 4.2. It show the degree of membership with which a car speed belongs to subset 'high'. Full membership of the class is represented by membership of 1, while 0 represents no membership. At speed of 60km/h the car speed doesn't belong to

subset of high speed at 120km/h and above the speed fully belongs to high speed. At speed between 60-120 km/h the membership increases linearly between 0 and 1. The degree that the speed belongs to subset high is called membership value.

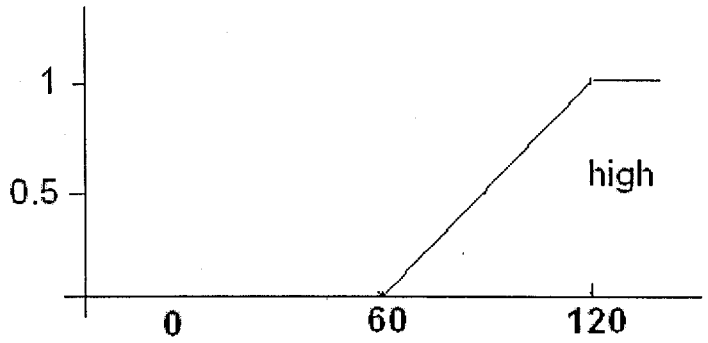

Figure 4.2 High speed subset

Normally fuzzy concepts have a number of values to describe the various ranges of values of the objective term that they describe. For example, the fuzzy concept 'speed' may have the values 'High', 'Medium' and 'Slow'. Typically, the membership functions of these values are as shown in the Fig. 4.3.

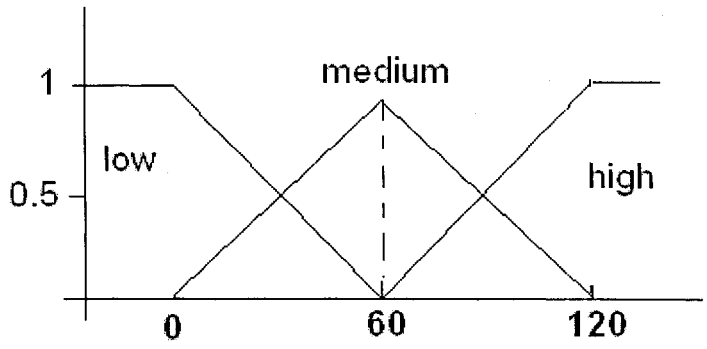

Figure 4.3 Speed in km/h

We can extend the above values by adding very low and very high. The real power of fuzzy logic systems, compared to crisp logic systems, lies in the ability to represent a concept using a small number of fuzzy values.

Similar procedure is followed in dividing subsets of the distance between the two cars as shown in Fig. 4.4, and the braking pressure in Fig. 4.5.

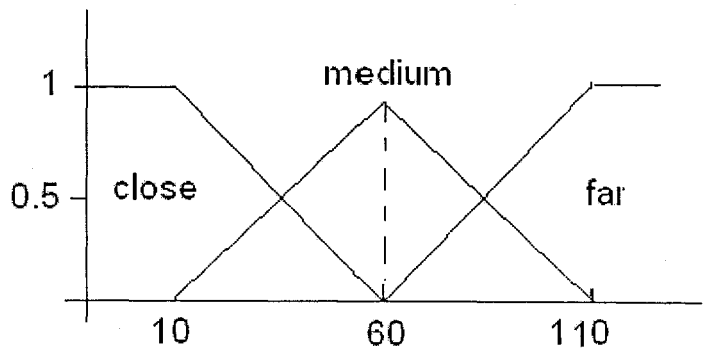

Figure 4.4 Distances in meters

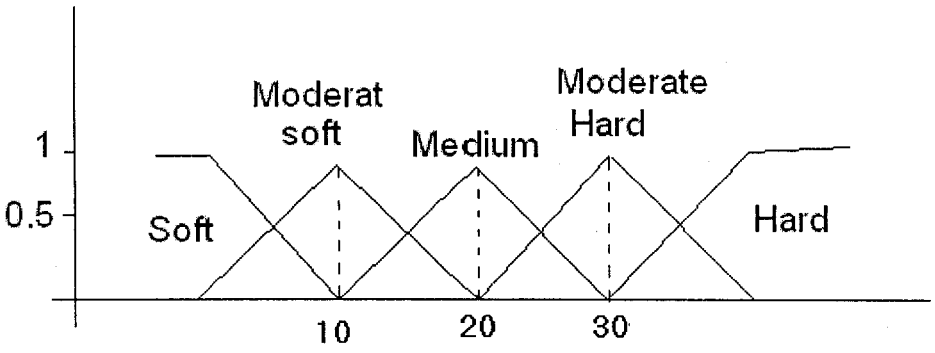

Figure 4.5 Braking force in Pa

Step3: fuzzify the input or determine the grade of each antecedent part to evaluate to what degree does each input belong to each of the appropriate input fuzzy sets. Therefore, the first part of the development of a fuzzy system is to develop the membership functions to translate each "crisp" input, which can be any number depending on the input signal, to a "fuzzified" variable which is contained between 0 and 1.

Consider a speed of 65 km/h and a distance of 100 meters. The membership of speed of 65 km/h on the medium subset is 0.9 and 0.1 in the high subset As depicted in Fig 4.6. The membership of 100 meter is 0.7 in the 'far' subset and 0.3 in the medium subset as in fig 4.7.

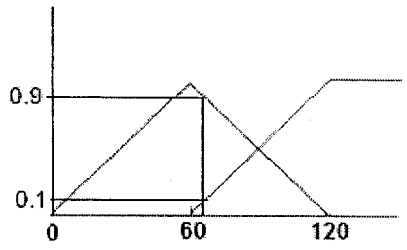

Figure 4.6 Membership of 65 km/h speed

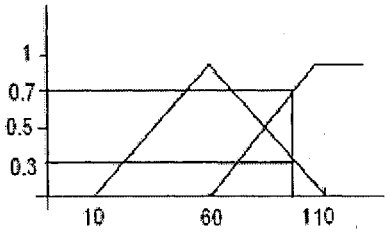

Figure 4.7 Membership of 100 m distance

The four rules are

1. MS (0.9) and MD (0.3) then MB (0.3)
2. HS (0.1) and LD (0.7) then MSB (0.1)
3. MS (0.9) and LD (0.7) then MSB (0.7)
4. HS (0.1) and MD (0.3) then MHB (0.1)

Since this is an AND operation, the minimum criterion is used, and the fuzzy set Medium of the variable "Brake Pressure" is cut at 0.3 and the patches are shaded up to that area. This is illustrated in the Fig. 4.8 and 4.9 below.

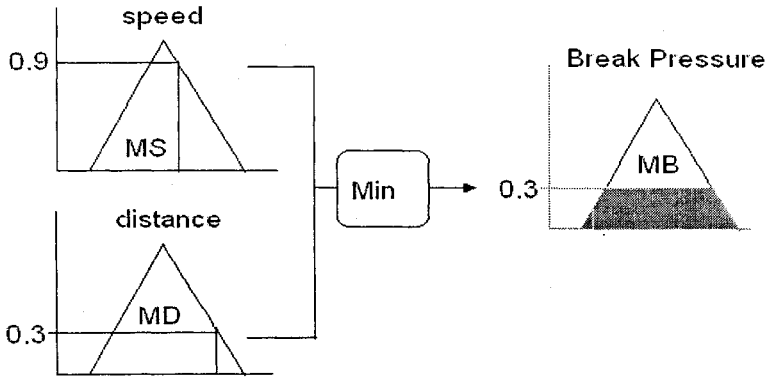

Figure 4.8 Fuzzy if-then of the first rule

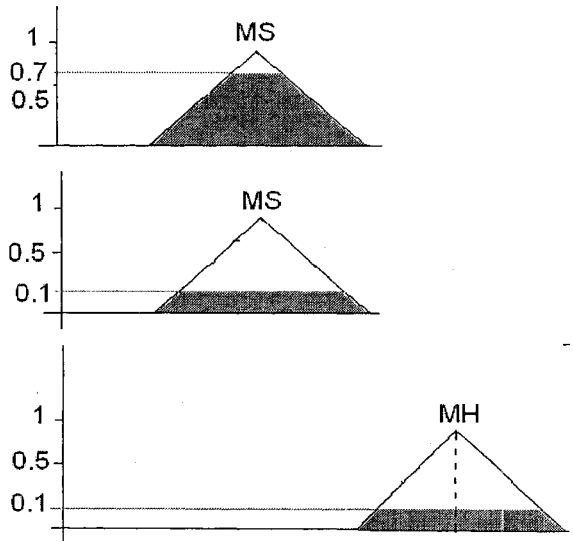

Figure 4.9 Fuzzy if-then of the 3 rules 2

The four results overlaps and is reduced to Fig. 4.9

Step 4: defuzzification

The result of the fuzzy controller is a fuzzy set (of Break Pressure). In order to choose an appropriate representative value as the final output (crisp values), defuzzification must be done. There are numerous defuzzification methods, but the most common one used is the center of gravity of the set as shown in Fig. 4.10.

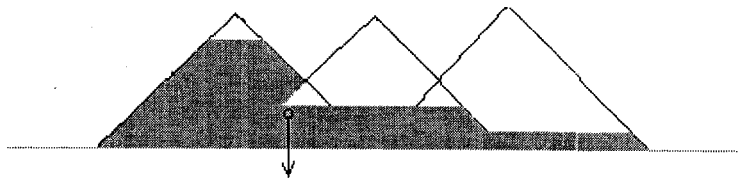

Figure 4.10 Defuzzification step

4.5 Neuro-Fuzzy Networks

A neuro-fuzzy network is a fuzzy system that uses a learning algorithm derived from neural network theory to determine its parameters. This s accomplished by processing data samples and using fuzzy rules and fuzzy reasoning as a basis for its training.

A neuro-fuzzy system can be viewed as a 3-layer feedforward neural network. The first layer represents input variables, the middle (hidden) layer represents fuzzy rules and the third layer represents output variables. Fuzzy sets are encoded as (fuzzy) connection weights. It is not necessary to represent a fuzzy system like this to apply a learning algorithm to it. However, it can be convenient, because it represents the data flow of input processing and learning within the model.

In this research we use neuro-fuzzy structure to approximate the location of landmarks in the x-ray. A general block diagram of the proposed machine is shown in Fig. 4.11. It is a non-linear mapping structure with multiple-input multiple-output features. The non-linear mapping from $\{x_1,...,x_n\}$ to $\{y_1,...,y_m\}$ is realized by a set of fuzzy inference rules. The inputs to the network are the features extracted for the x-ray images such as size, rotation angles, and shift of different point. The outputs are the estimated locations of the landmarks.

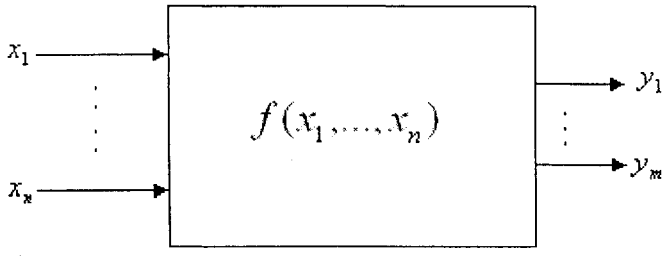

Figure 4.11 A block diagram of the neuro-fuzzy network

This non-linear mapping system is a supervised system. A system is said to be supervised if it is trained by a set of patterns with known output and then asked

to predict the output of unknown pattern based on the experience acquired during the training. In our case the expected output are the locations of the landmarks, which are obtained by the expert for the set of training x-ray. Considering the features of the system, we use simplified inference rules for which the consequence parts are real numbers. In general, the inference rule set for proposed neuro-fuzzy system is as follows [41]:

$R^{(1)}$: **IF** x_1 *is* A_{11} *and* x_2 *is* A_{21},...,*and* x_n A_{n1} **THEN** y_1 *is* y_{11},...,*and* y_m *is* y_{1m}

$R^{(2)}$:**IF** x_1 *is* A_{12} *and* x_2 *is* A_{22},...,*and* x_n A_{n2} **THEN** y_1 *is* y_{12},...,*and* y_m *is* y_{2m}

...

$R^{(k)}$: **IF** x_1 *is* A_{1k} *and* x_2 *is* A_{2k},...,*and* x_n A_{nk} **THEN** y_1 *is* y_{1k},...,*and* y_m *is* y_{km}

where $R^{(i)}$ is the i*th* fuzzy rule, $A_{1j}, A_{2j},..., A_{nj}$ (j=1,...,k) are fuzzy subsets of each of the *n* input features. $y_{j1},...,y_{jp}$ (j=1,...,k) are real numbers in [0,1]. The membership function A_{ij} (I=1,...,n; j=1,..k) can be any membership function outlined in Chapter 3.

When inputs to the system $x_1 = x_1^o$ and $x_2 = x_2^o ... x_n = x_n^o$ the output will be $y_1 = y_1^o$ and $y_2 = y_2^o ... y_p = y_p^o$, where $x_1^o, x_2^o,..., x_n^o$ and $y_1^o, y_2^o,..., y_p^o$ are real numbers.

In this research a generalized bell membership function (GBMF) is used in the antecedent part with a single tone in the consequence part.

The membership functions are arranged to cover all the fluctuation of the input feature as shown in Fig. 4.12. S_n is the number of membership functions or the number of subsets assigned for fluctuation range. $x_n^{min} \le x_n \le x_n^{max}$. It should be noted that the membership function in the antecedent are complementary to each other. Each input signal x_i activates only two neighboring membership

functions and the sum of the grades of the membership functions is always equal to 1.0.

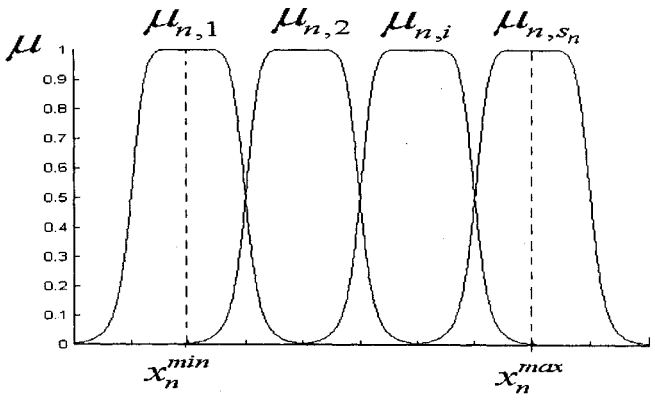

Figure 4.12 Complementary membership functions

The matching grade of an antecedent part of rule $R^{(i)}$ for an input $\{x_1,...,x_n\}$ becomes

$$\mu_R{}^{(i)}(x_1,x_2,...,x_n) = \mu_{A_1^{(i)}}(x_1) \otimes \mu_{A_2^{(i)}}(x_2) \otimes ... \otimes \mu_{A_n^{(i)}}(x_n) \tag{4.5}$$

where $\otimes$ is an AND operator, for which an algebraic product is employed here as AND operator. Taking the center of gravity as defuzzification, the inference result for the output y_j becomes

$$y_j = \frac{\sum_{i=1}^{N} \mu_{R^{(i)}} w_j^i}{\sum_{i=1}^{N} \mu_{R^{(i)}}} \tag{4.6}$$

where N is the number of rules in a rule set. The weight w_j^i is an adaptive parameter between nodes *i* and *j*, which is determined through an iterative learning process to train the network to do a desired action [42]. y_j is the crisp output of the network representing the estimated location of the sought landmark.

4.6 Structures and Algorithms of Neuro-Fuzzy Network

The proposed neuro-fuzzy network is implemented in layered feedforward network structures as shown in Fig. 4.13. The network has n inputs $\{x_1,...,x_n\}$ and a single output. This is not a limitation, since a fuzzy algorithm with n inputs and m outputs is equivalent to m fuzzy algorithms with n inputs and one output. The network adopted to model this fuzzy system has a three-layer feed-forward structure also used in [43-45]. Different types of fuzzy neurons are used in different layers. Each layer has a different function from the other layers.

The first layer is the fuzzification layer. It accepts the data into the network and calculates the degree of membership of each input vector with respect to fuzzy subsets $A_{1j}, A_{2j},..., A_{Nj}$ (j=1,2...K). This layer is composed of groups of n neurons. The ith neuron in the jth group will compute the membership of input x_j to the subset A_{ji} for the ith fuzzy term in the jth rule.

The second layer performs the inference of rules by means of the product operator. The number of nodes in this layer is equal to the number of rules in the fuzzy system. The output of neuron j in this layer is

$$R^{(j)} = \prod_{i=1}^{n} \mu_{ij}(x_i) \tag{4.7}$$

Equation 4.7 corresponds to the activation strength of the jth rule for input $x_i (i = 1,...,N)$. Note that neurons in this layer are fixed, as no modifiable parameter is associated with them. The structure of the network is completely defined once the size of first and second layers is determined.

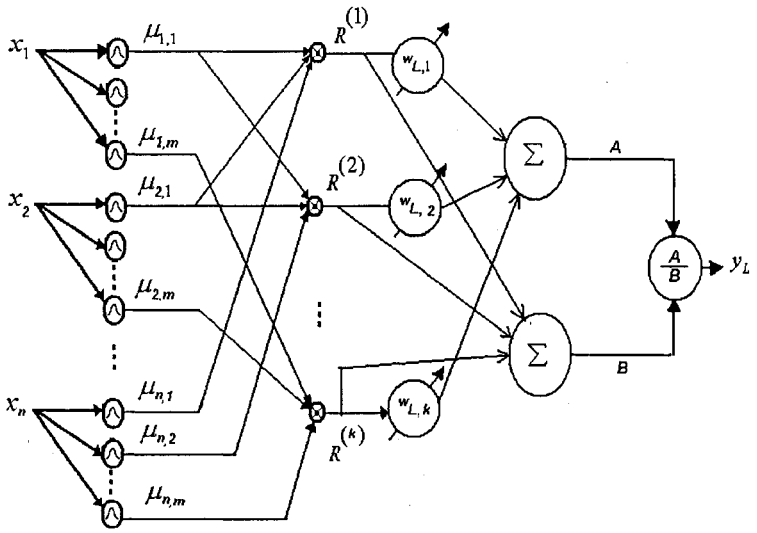

Figure 4.13 The architecture of the proposed neuro-fuzzy network

The third layer performs the summation of the rules multiplied by the weights and the sum of the weights as follows:

$$A = \sum_{k=1}^{nr} R^{(k)} w_{Lk} \tag{4.8}$$

$$B = \sum_{k=1}^{nr} w_{Lk} \tag{4.9}$$

Where *nr* is the number of rules.

The fourth layer performs the defuzzification phase by means of the weighted sum method. The output of the single unit in this layer is:

$$y_L = \frac{A}{B} \tag{4.10}$$

4.7 Learning Algorithm

Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly, then the training, or learning, begins.
We will be using supervised training. Supervised training involves a mechanism of providing the network with the desired output either by manually "grading" the network's performance or by providing the desired outputs with the inputs

Both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights that control the network. This process occurs over and over as the weights are continually tweaked. The set of data that enables the training is called the "training set." or "prototypes". During the training of a network the same set of data is processed many times as the connection weights are refined.

The convergence of training algorithm measures the ability to predict the right answer. The training is stopped when the system reaches some statistically desired point, or accuracy. However, some networks never learn and the desired point will never be reached. This could be because the input data does not contain the specific information from which the desired output is derived. Networks also don't converge if there is not enough data to enable complete learning. Ideally, there should be enough data so that part of the data can be held back as a test. Many layered networks with multiple nodes are capable of memorizing data. To monitor the network to determine if the system is simply memorizing its data in some significant way, supervised training needs to hold back a set of data to be used to test the system after it has undergone its training. If the network is trained using all the data and tested using the same set of data, the network is said to have memorized the data.

Let us consider a smaller network with two inputs x_1 and x_2, one output y and two fuzzy subsets for each input producing four rules as shown in Fig 4.13. The two fuzzy sets are shown in fig 4.14.

x_1 and x_2 are presented as inputs to the network. Each neuron in the first layer represents a fuzzy subset. The output of the first layer is the membership of each input to one of the fuzzy subsets as follows:

The output of the first fuzzy neuron is $\mu_{A_{11}}(x_1)$ which represents the membership of input variable x_1 to the fuzzy subset A_{11}.

The output of the second fuzzy neuron is $\mu_{A_{12}}(x_1)$ which represents the membership of input variable x_1 to the fuzzy subset A_{12}. The same applies for the rest of the neurons.

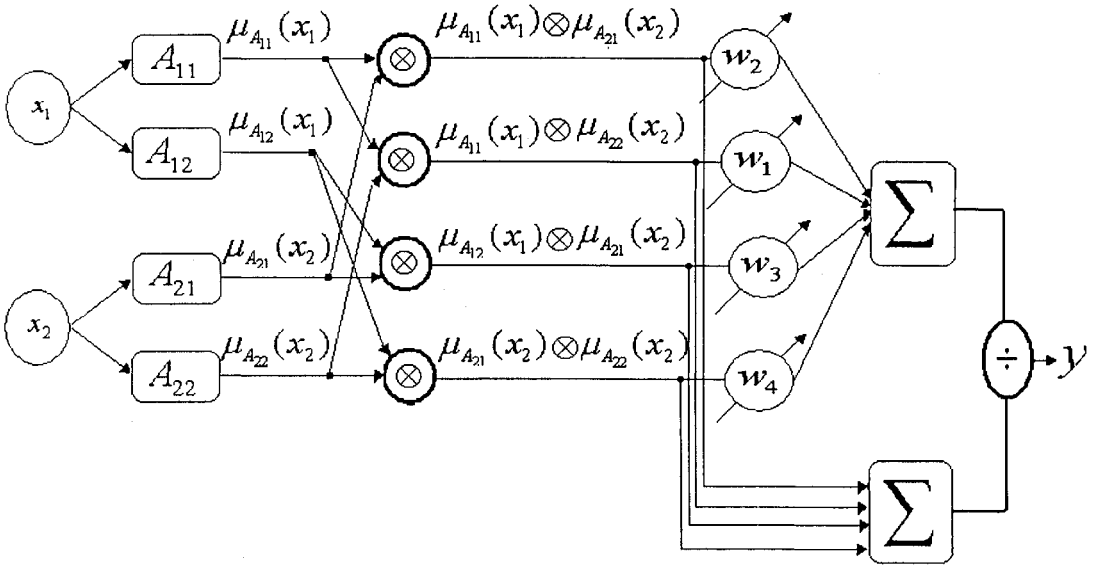

Figure 4.14 A simplified neuro-fuzzy network with two inputs and one output

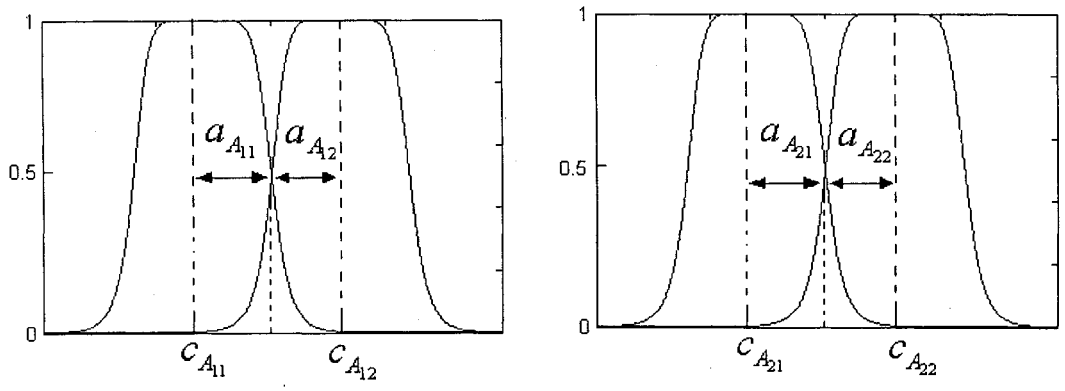

Figure 4.15 Two fuzzy subsets

$$\mu_{A_{11}}(x_1) = \frac{1}{1+\left|\frac{x_1 - c_{A_{11}}}{a_{A_{11}}}\right|^{2b_{A_{11}}}} \quad , \quad \mu_{A_{12}}(x_1) = \frac{1}{1+\left|\frac{x_1 - c_{A_{12}}}{a_{A_{12}}}\right|^{2b_{A_{12}}}}$$

$$\mu_{A_{21}}(x_2) = \frac{1}{1+\left|\frac{x_1 - c_{A_{21}}}{a_{A_{21}}}\right|^{2b_{A_{21}}}} \quad , \quad \mu_{A_{22}}(x_2) = \frac{1}{1+\left|\frac{x_1 - c_{A_{22}}}{a_{A_{22}}}\right|^{2b_{A_{22}}}} \tag{4.11}$$

The output of the second layer is the product of the outputs of each two neurons in the first layer. This will produce four rules since we have two sets and two input variables. Each neuron in this layer will be responsible for producing the if part of one rule as follows:

$$R^{(1)} = \mu_{A_{11}}(x_1) \otimes \mu_{A_{21}}(x_2)$$

$$R^{(2)} = \mu_{A_{12}}(x_1) \otimes \mu_{A_{21}}(x_2)$$

$$R^{(3)} = \mu_{A_{11}}(x_1) \otimes \mu_{A_{22}}(x_2)$$

$$R^{(4)} = \mu_{A_{12}}(x_1) \otimes \mu_{A_{22}}(x_2) \tag{4.12}$$

The output of the third layer is the sum of all rules after multiplying each rule by a weight w_i. The values of the weight are to be determined by the training algorithm. The output of third layer is

$$y = \frac{\sum_{i=1}^{4} R^{(i)}.w_i}{\sum_{i=1}^{4} R^{(i)}} \tag{4.13}$$

or

$$y = \frac{R^{(1)}.w_1 + R^{(2)}.w_2 + R^{(3)}.w_3 + R^{(4)}.w_4}{R^{(1)} + R^{(2)} + R^{(3)} + R^{(4)}} \tag{4.14}$$

To obtain the weights that best represent the input-output data we minimize an error function, which is a measure of the difference between the desired output and the obtained output as follows:

$$E = \frac{1}{2}(y^{rp} - y^{p})^2 \tag{4.15}$$

Where y^{rp} is the target output for the pth input data $\{x_1^p,...,x_n^p\}$ and y^p is the calculated output of the system corresponding to the same input data. Using input-output data and gradient decent algorithms, we optimize the weights by minimizing the objective function E given by equation (4.15).

The gradient of E is defined as follows:

$$\nabla E(w) = [\frac{\partial E(w)}{\partial w_1}, \frac{\partial E(w)}{\partial w_2}, ..., \frac{\partial E(w)}{\partial w_n}]^T \tag{4.16}$$

We would like to change the gradient in the direction that minimizes the error function such that

$$w_{next} = w_{now} - \eta \frac{\partial E}{\partial w} \tag{4.17}$$

or

$$w_i(t+1) = w_i(t) - \eta_w \frac{\partial E}{\partial w_i} \tag{4.18}$$

where t is the number of iterations of learning and η_w is a positive step size regulating to what extent to proceed in that direction.

Ideally, we wish to obtain a value of w_{next} that satisfies the following:

$$E(w_{next}) = \frac{\partial E(w)}{\partial w}\bigg|_{w=w_{next}} = 0 \tag{4.19}$$

In practice, it is difficult to solve Equation (4.19) analytically. For minimizing the objective function, the procedure is repeated until the error reaches a sufficiently small value.

$$\frac{\partial E}{\partial w} = (y^{rp} - y^p)\frac{dy^p}{dw} \tag{4.20}$$

and

$$\frac{dy^p}{dw_i} = \frac{R^{(i)p}}{\sum_{i=1}^{n} R^{(i)p}} \tag{4.21}$$

where $R^{(i)p}$ is the membership value of the *i*th rule corresponding to the *p*th input-output example.

w_i can be updated as follows

$$w_i(t+1) = w_i(t) - \eta_i \frac{R^{(i)p}}{\sum_{i=1}^{n} R^{(i)p}} .(y^p - y^{rp}) \tag{4.22}$$

Where *t* is the learning cycle.

Using the input-output example with learning rules of equation (4.22) repeatedly, the weights are updated to minimize the error function, ultimately reaching global minimum since $\frac{\partial^2 E}{\partial w_i^2} \geq 0$ can be obtained for all rules.

Using gradient decent, the membership function of the rules may be tuned to better reflect the problem at hand. Consider the symmetric (GBMF) shown in fig 4.14 for the jth antecedent in the ith rule. Such a function can be represented by the width of the membership function a_{ij} and the slope of the curve at the cross over points b_{ij}, and therefore the entire rules of equation (4.16) can be parameterized through b_{ij}, a_{ij} and w_i [46].

The parameter a_{ij} is updated as follows:

$$a_{ij}(t+1) = a_{ij}(t) - \frac{\eta_a}{ns} . \frac{\partial E}{\partial a_{ij}} \tag{4.23}$$

Where η_a is the gradient-decent speed a_{ij}, *ns* is the number of samples used for training and

$$\frac{\partial E}{\partial a_{ij}} = \left(\frac{\partial E}{\partial y}\right) \cdot \left(\frac{\partial y^p}{\partial R^{(i)}}\right) \cdot \left(\frac{\partial R^{(i)}}{\partial \mu_{ij}(x_i)}\right) \cdot \left(\frac{\mu_{ij}(x_i)}{\partial a_{ij}}\right) \tag{4.24}$$

Similarly b_{ij} is upgraded as follows:

$$b_{ij}(t+1) = b_{ij}(t) - \frac{\eta_b}{ns} . \frac{\partial E}{\partial b_{ij}} \tag{4.25}$$

Where η_b is the gradient-decent speed a_{ij}, *ns* is the number of samples used for training and

$$\frac{\partial E}{\partial b_{ij}} = \left(\frac{\partial E}{\partial y}\right) \cdot \left(\frac{\partial y^p}{\partial R^{(i)}}\right) \cdot \left(\frac{\partial R^{(i)}}{\partial \mu_{ij}(x_i)}\right) \cdot \left(\frac{\mu_{ij}(x_i)}{\partial b_{ij}}\right) \tag{4. 26}$$

The value of c_{ij} in Fig.4.12 is computed as follows

$$c_i = \min_i + a_{i1} \quad \text{or} \quad c_i = \max_i - a_{i2} \tag{4.27}$$

where *i* is the feature number.

After training the GBMF can have the shapes similar to that shown in figure 4.13

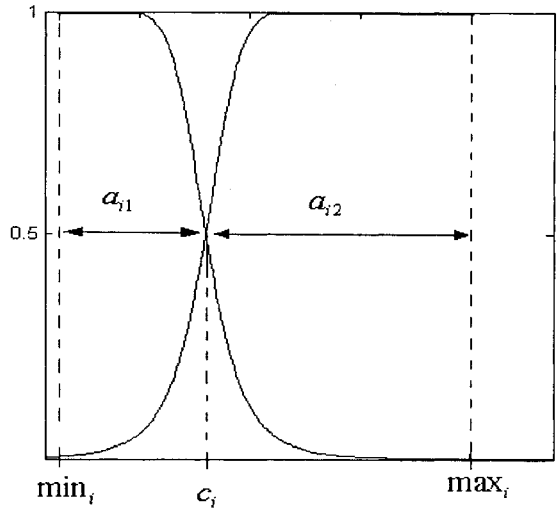

Figure 4.16 GBMF for the *ith* feature after training

4.8 Conclusions

In this chapter we have introduced the basic structure of fuzzy neurons and the use fuzzy system to estimate the expected outcomes based on the use of inference *if-then* fuzzy rules. We have also shown how problems are solved using neuro-fuzzy structure by mapping the input linguistic variables into a fuzzy set. After the problem at hand is formulated using fuzzy sets and fuzzy rules, the neuro-fuzzy network is trained and implemented using backpropagation techniques based on gradient decent method for minimizing and objective function. When the network is trained using a set of labeled data, it can be used to predict the outcome of an input new to the network.

Chapter 5

Features Extraction and Template Matching

5.1 Introduction

An image can be represented as a function, *f(x,y),* of two continuous variables representing the horizontal and vertical coordinates of a point in space. For monochrome images, *f(x,y)* is a scalar function represent the intensity of the point. For color images, *f(x,y)* is a vector function that denotes the color of the point (e.g. it could have three values for red, green and blue in the RGB coordinate system). Monochrome images are displayed using black and white and shades of gray, thus they are called "grayscale" images. A sampled grayscale image can be represented as a two-dimensional array of number of the form :

$$f(m,n) = f(1{:}m,1{:}n) \; for \; 1 \le m \le M \; and \; 1 \le n \le N$$

Grayscale images show the intensity of each pixel, which must always be finite and non-negative: $0 \le f(m,n) \le fmax$ .Grayscale images commonly use eight or sixteen bit integer representation. With 8-bit images the

maximum intensity value is $Fmax=2^8 - 1 = 255$. Pixels with intensity values around 255 are white and pixels with values near 0 are black. In this research we are dealing with 8 bit gray-scale images of size ranging between 461x461 to 563x563 pixels.

Since our images represent biological shapes, they cannot be easily described by a simple, scaled, rotated and shifted pattern that can be recognized by simple algorithms. There exists an infinite number of scales, rotations and deformations for each feature in the image. To extract features from an image we must detect the outer edges and make the measurements between the points of interest. Based on the extracted features, all images are clustered into groups of similar features. From each group one image is chosen as a prototype representing that group in the training set. Before features can be extracted from the image, several image-processing techniques need to be applied in order to be able to extract the features. We first enhance the appearance of the image using histogram equalization then detect the outer edges on the enhanced image. On the outer edges we chose four points to make the measurements and extract the features.

5.2 Image Enhancement

Image enhancement refers to any technique that improves or modifies the image data, either for purposes of visual evaluation or for numerical processing. Image enhancement techniques include gray level and contrast manipulation, noise reduction, edge sharpening, and image filtering. Image enhancement techniques can be categorized into point and region-based operations. Point operations modify pixels of an image based on the value of the pixel while region-based operations calculate a new pixel value based on the values in the local neighborhood. We utilize the histogram equalization to enhance the appearance of the image

5.2.1 Image Histogram

The image histogram is an estimate of the probability density of the image pixels. It measures the frequency of occurrence of the pixel luminance values. Many higher-level image processing tasks require the calculation of a histogram. Fig 5.1 shows a histogram of an x-ray image. This histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. For an 8-bit grayscale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those grayscale values. Form fig 5.1 we see that the pixels are normally distributed. Most of the pixels have values of 0 to 150.

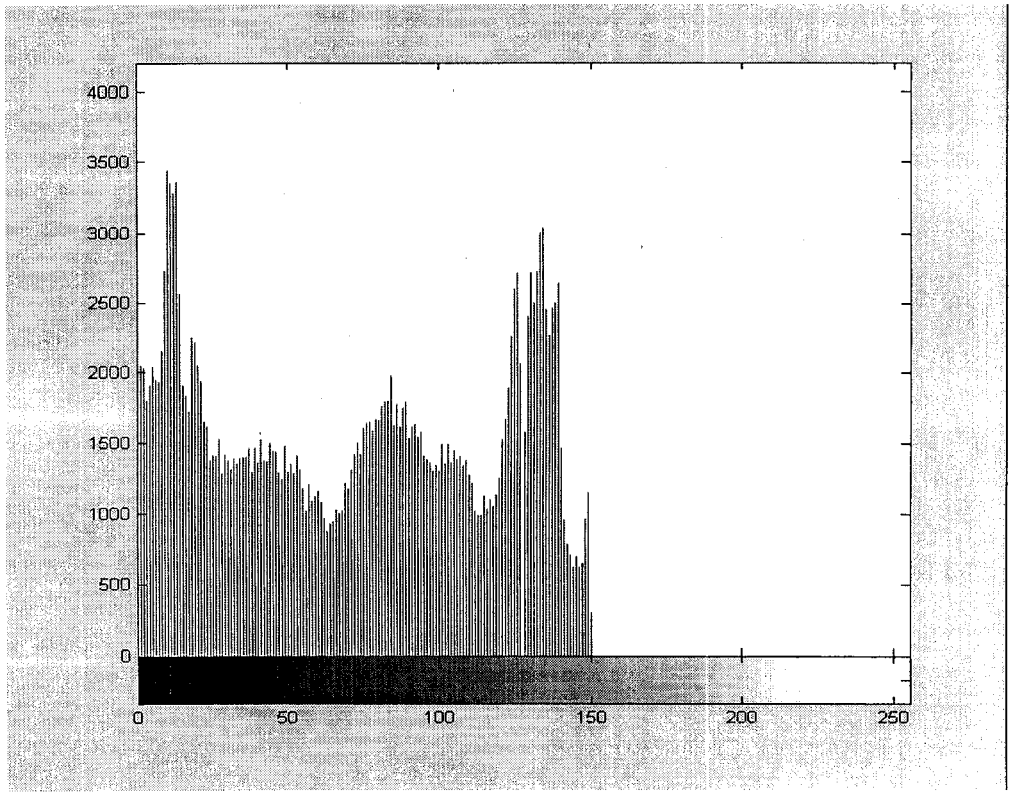

Figure 5.1 X-ray Image histogram

5.2.2 Histogram Equalization

Histogram equalization is as a contrast enhancement technique which aimes to obtain a new enhanced image with a uniform histogram. This can be achieved by using the normalized cumulative histogram as the grey scale mapping function. Pixels gray values will be mapped to new values but the actual number of intensities in the resulting image will be equal or less than the original number of intensities. Histogram modification can be performed as follows:

1. Compute histogram: The first step is accomplished by counting each distinct pixel value in the image. Since our images are 8-bit pixels, the size of the histogram array is 256 (0-255). For each element corresponding to one level, we determine as how many pixels have that level.
2. Calculate the normalized sum of the histogram: this step requires another array to store the sum of histogram values. In this array element, 255 will contain the sum of histogram elements 255,254,253....0 and element 1 will contain the sum of histogram element 1 and 0. Each element in this array is normalized by multiplying it by maximum pixel value/number of pixels. For an 8-bit image of size 512x512 this factor will be 255/ 262144
3. The result from step 2 yields a Look Up Table (LUT) that will be used to map the original image. This LUT can be used to map the original image to a new image with a more equalized histogram than that of the input image.

Fig 5.2 show the new intensity distribution of fig 5.1 while Fig. 5.3 and 5.4 show the image before and after applying histogram equalization.

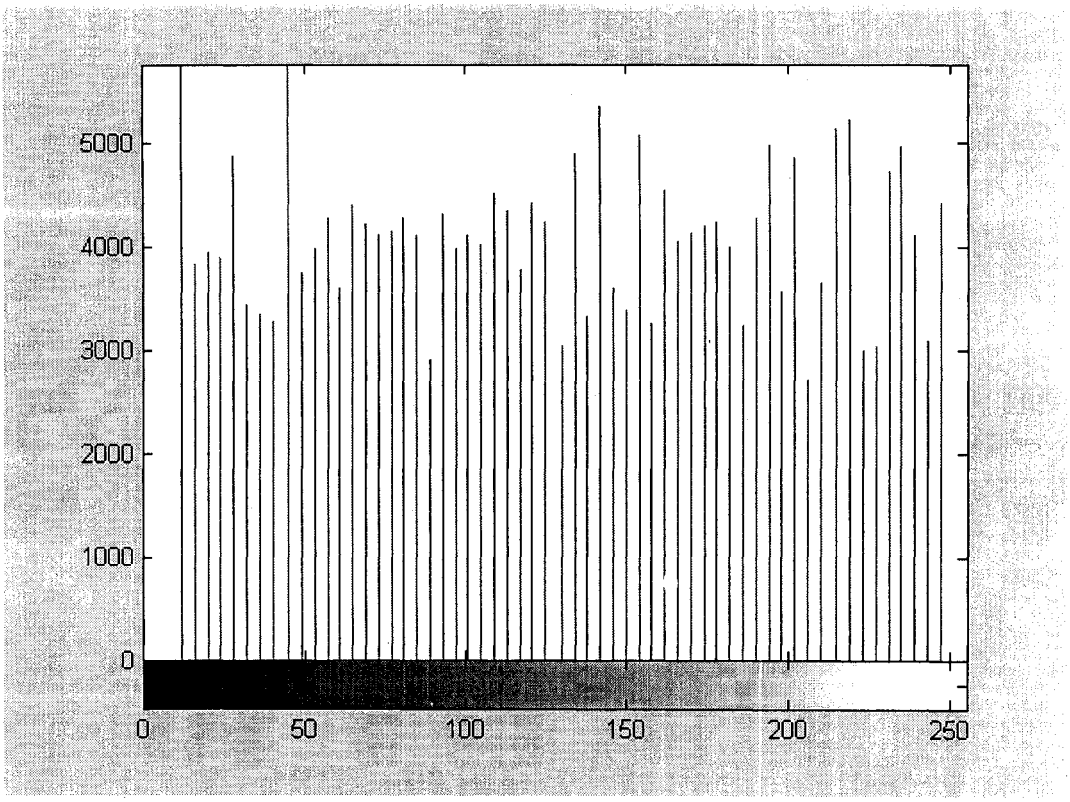

Figure 5.2 Equalized histogram of the image

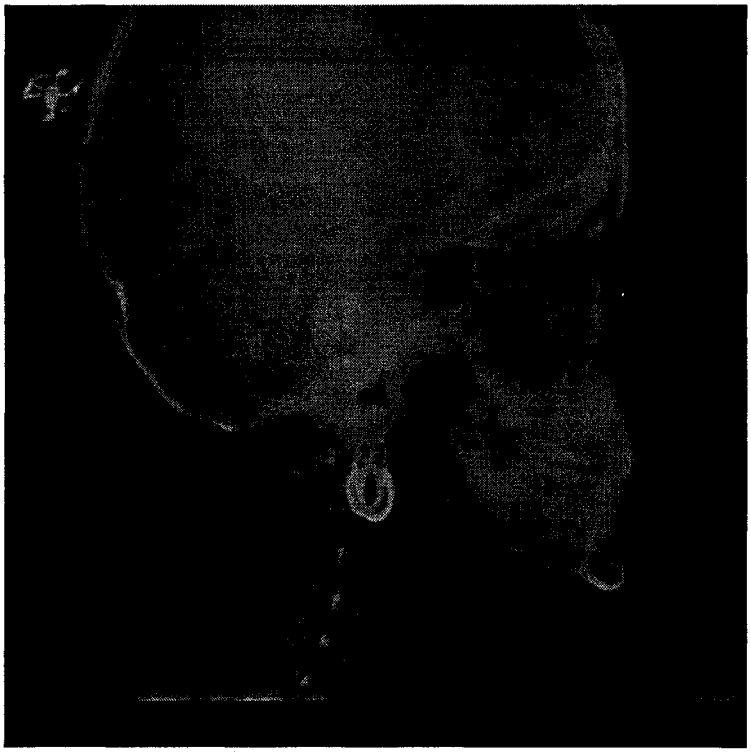

Figure 5.3 Image before applying histogram equalization

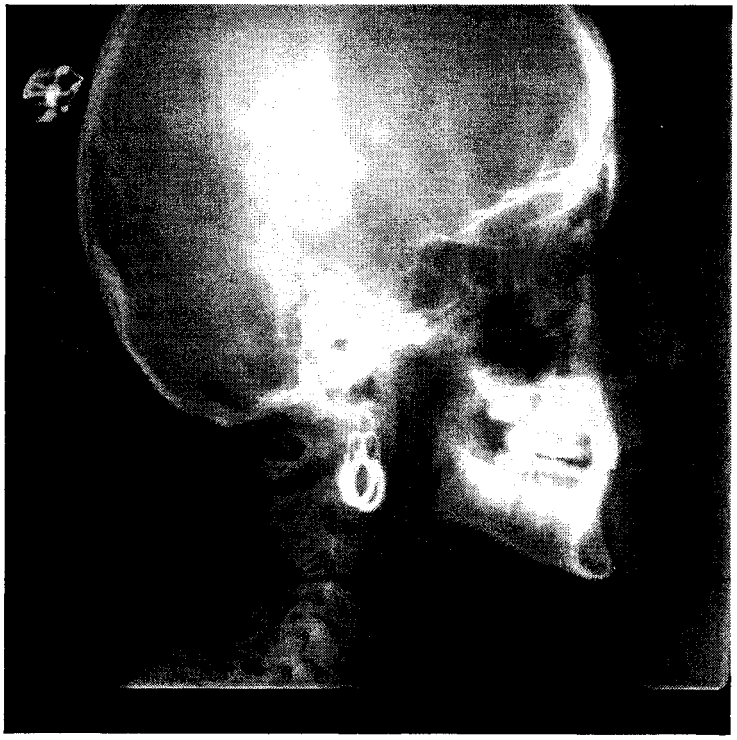

Figure 5.4 Image after applying histogram equalization

5.3 Edge Detection

An edge is the boundary between two regions with relatively distinct gray-level properties. The idea underlying most edge-detection techniques is the computation of a relative derivative operator. Fig. 5.5 illustrates this concept. The picture on the left shows an image of a light stripe on a dark background, the gray-level profile of a horizontal scan line, and the first and second derivatives of the profile. The first derivative is positive when the image changes from dark to light and zero where the image is constant. The second derivative is positive for the part of the transition associated with the dark side of the edge and negative for the transition associated with the light side of the edge. Thus the magnitude of the first derivative can be used to detect the presence of an edge in the image and the sign of the second derivative can be used to detect whether a pixel lies on the light or dark side of the edge.

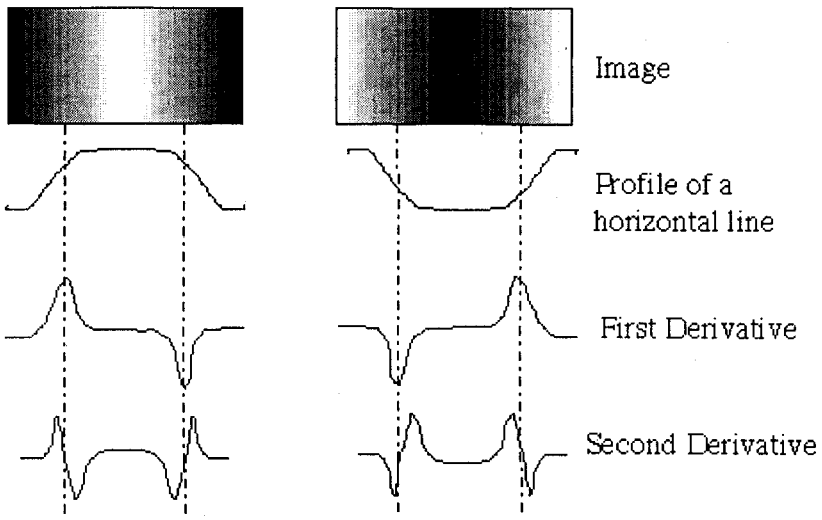

Figure 5.5 First and second derivative of positive and negative edges

5.3.1 Sobel Operator

We will use two-dimensional spatial masks known as Sobel operator to find lines of different directions in an image. The mask is centered upon each pixel in the

image and the following weighted calculation is performed (*x* is the original image pixels, y_1 is the output using *h* mask and y_2 is the output using v mask and)

$$y_1(i,j) = h_1.I(i-1,j-1) + h_2.I(i-1,j) + h_3.I(i-1,j+1) + h_4.I(i,j-1) + h_5.I(i,j) + h_6.I(i,j+1) + h_7.I(i+1,j-1) + h_8.I(i+1,j) + h_9.I(i+1,j+1) \quad (5.1)$$

$$y_2(i,j) = v_1.I(i-1,j-1) + v_2.I(i-1,j) + v_3.I(i-1,j+1) + v_4.I(i,j-1) + v_5.I(i,j) + v_6.I(i,j+1) + v_7.I(i+1,j-1) + v_8.I(i+1,j) + v_9.I(i+1,j+1) \quad (5.2)$$

Masks:

h

1	0	-1
2	0	-2
1	0	-1

-1	-2	-1
0	0	0
1	2	1

The masks shown above respond more strongly to lines of different directions in the image. The first mask would respond very strongly to vertical lines with a thickness of one pixel. The maximum response would occur when line passed through the column row of the mask. The direction of a mask can be established by noting that the preferred direction is weighted with a larger coefficient that the other possible directions. The edge strength of the image can be obtained using the following nonlinear operator. [48]

$$y(i,j) = \sqrt{y_1(i,j)^2 + y_2(i,j)^2} \quad (5.3)$$

Figure 5.6 shows the original image after applying histogram equalization; Fig. 5.7 shows the image after applying Sobel operator.

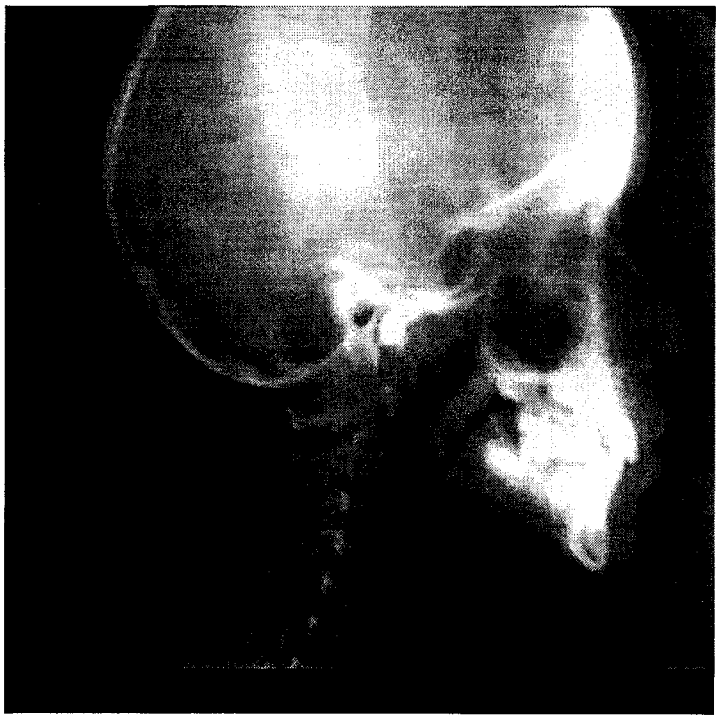

Figure 5.6 Original image

Figure 5.7 Image after applying Sobel operator

5.3.2 Median Filtering

Median filtering is a non-linear operation used for smoothing the image and removing the noise. In median filtering, the gray level of each pixel is replaced by the median of the gray levels in a neighborhood of that pixel, instead of by the average. In order to perform median filtering in a neighborhood of a pixel, we first sort the values of the pixel and its neighbors, determine the median, and assign this value to the pixel. The size of the sorting neighborhood is controlled by the size of the filter. If the window size is too large or too small, pixels are forced to be similar to a different neighborhood. We chose a 5x5 median filter as shown in Fig. 5.8. Fig. 5.10 show image converted to binary image.

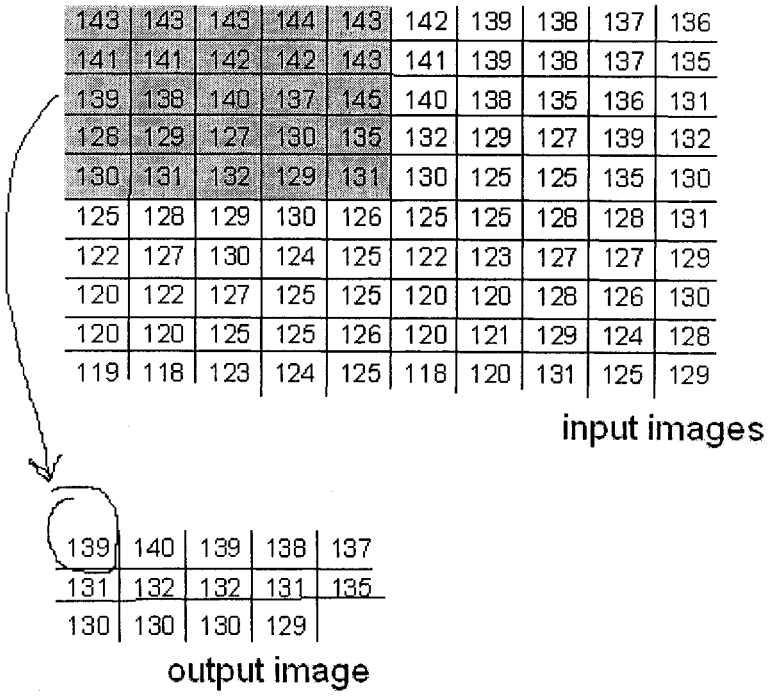

Figure 5.8 Median filtering

Figure 5.9 Image after applying median filter of size 5x5

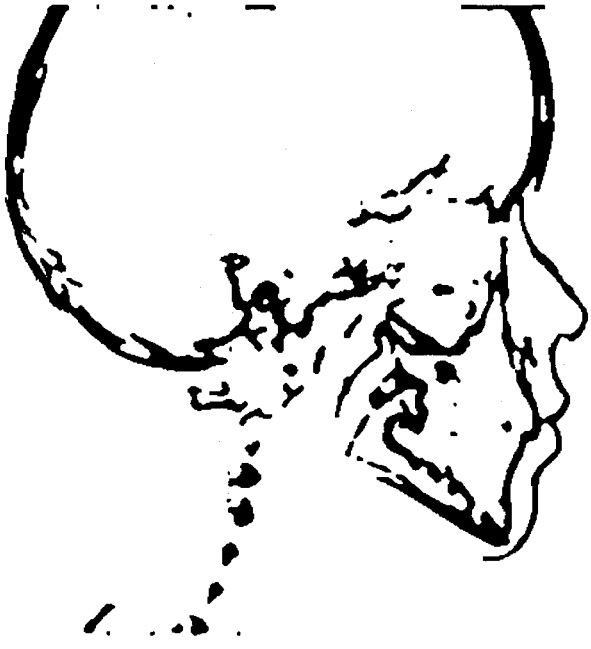

Figure 5.10 Image converted to binary

5.4 Features Extraction

After edge detection and noise removal, four points are located on the image as shown in Fig.5.11. The points are located by tracing the image in the directions shown in Fig. 5.10 until the first or last zero-value is found. Points (P1) and (P4) are located by tracing the image vertically from top to bottom starting from the left side for P1 and right side for P4. Point (P2) is found by extending a line from P1 until the last zero value is found. Point (P3) is found by scanning diagonally from right to left starting from the right hand corner. A shape similar to that shown in Fig 5.12 is constructed and more features are made using the extracted four points as listed in Table 5.1.

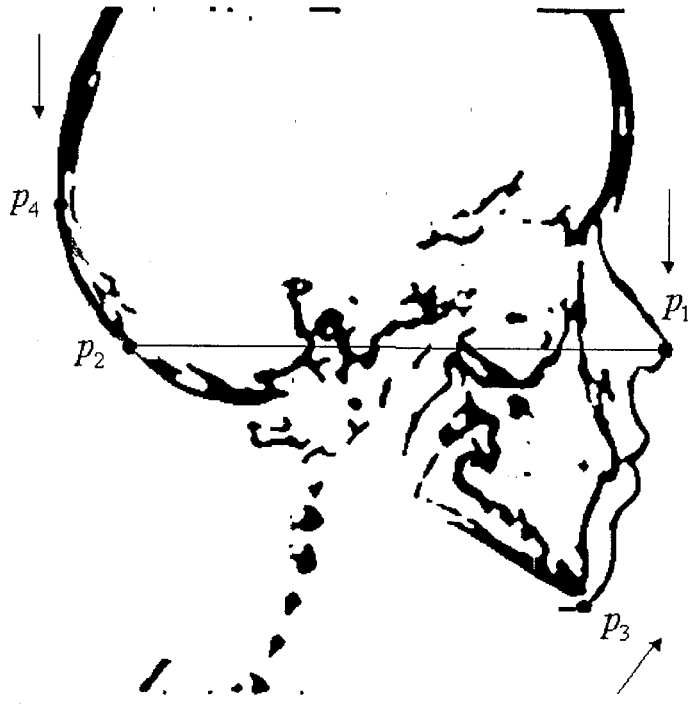

Figure 5.11 Detection of four points on the edges

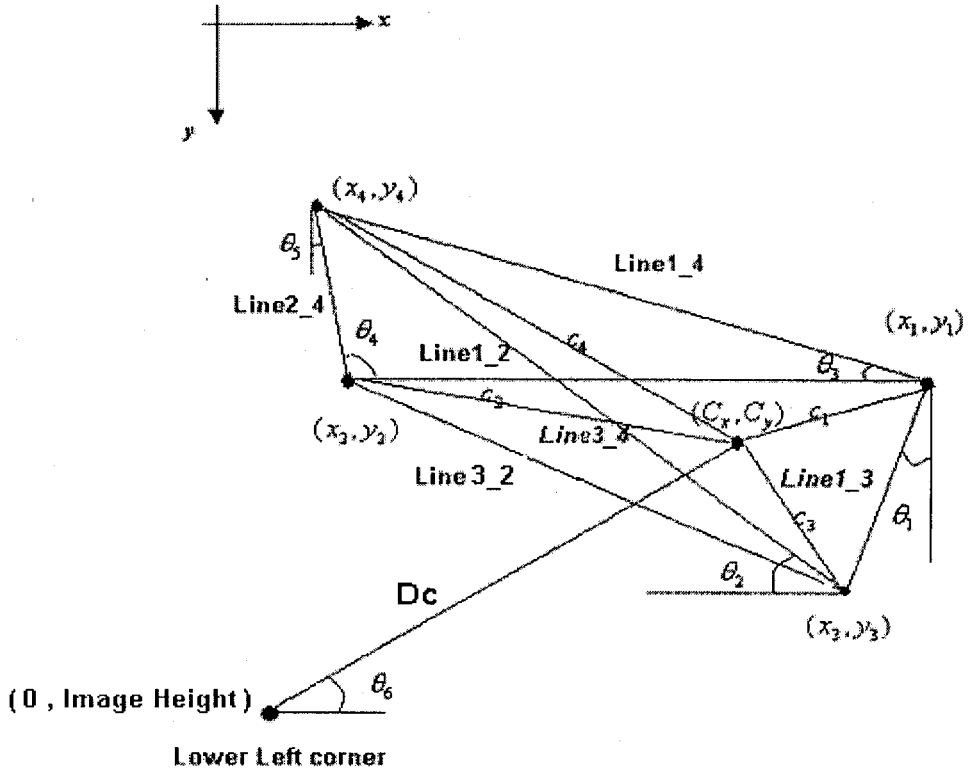

Figure 5.12 Features made from the extracted four points

Table 5.1 Name and description of the made features

No	Name	Description
1	Line1_2	Distance between P1 and P2
2	Line1_3	Distance between P1 and P3
3	Line1_4	Distance between P1 and P4
4	Line2_4	Distance between P2 and P4
5	Line2_3	Distance between P2 and P3
6	Line3_4	Distance between P3 and P4
7	c_x	Horizontal coordinate of the center of gravity
8	c_y	Vertical coordinate of the center of gravity
9	c_1	Distance between center of gravity and P1
10	c_2	Distance between center of gravity and P2
11	c_3	Distance between center of gravity and P3
12	c_4	Distance between center of gravity and P4
13	θ_1	Angle between Line1_3 and the vertical axis
14	θ_2	Angle between Line3_4 and the horizontal axis
15	θ_3	Angle between Line1_4 and the horizontal
16	θ_4	Angle between Line2_4 and Line1_2
17	θ_5	Angle between Line2_4 and the vertical axis.
18	Pr	Perimeter of the shape = Line1_3 + Line3_2 +Line2_4 + Line4_1
19	A	Area of the shape.
20	θ_6	Angle between the line connecting the center of gravity to the lower left corner and the horizontal
21	Dc	Distance between center of gravity and the lower left corner of the image

Euclidian distance is used for computing all distances. Distances were measured in pixels while angles are measured in degrees.

5.5 Data Clustering

Clustering is used to categorize images by partitioning them into several groups such that similarity within a group is larger than that among other groups. Since all clustering algorithms are sensitive to the range of elements in the input vectors, all features are normalized within the unit hypercube.

5.5.1 Subtractive Clustering

Subtractive clustering proposed by Chiu [49] is used to create initial clusters before applying the k-means, since k-means might not converge unless a good initial estimate of the clusters centers is provided. In subtractive clustering methods, all images are considered as candidates for cluster centers.

Consider a collection of *n* images each represented by vectors $\{X_0, X_1, \ldots, X_{n-1}\}$ of *m* dimensional space such that $X = \{x_0, x_1, \ldots, x_{m-1}\}$ where m is the number of features. Since each vector is a candidate for cluster centers, a density measure of vector X_i is defined as

$$D_i = \sum_{j=0}^{n-1} \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{\left(r_a/2\right)^2}\right) \tag{5.3}$$

where r_a is a positive constant defining the radius of the neighborhood. Contribution from data points outside this radius to the density measure is smaller.

The radius r_a is estimated as follows:

Let $d_i = \frac{\sum_{j=1}^{n} \left\|x_i - x_j\right\|}{n}$ be the average distances form vector i to all other vectors then r_a is defined as twice the minimum distance or $r_a = \min(d_i) * 2$.

After the density measure has been calculated for each image or data vector, the image with the highest density measure is selected as the center of the first cluster. Let x_{s1} be an image with density D_{S1}. Next, the density measure for all images is revised by the formula

$$D_i = D_i - D_{si} \exp(-\frac{\|x_i - x_{si}\|^2}{(r_b / 2)}) \quad (5.4)$$

where r_b is a positive number selected to be about 1.5 r_a to prevent closely spaced centers [49]. Revising the density using (5.4), the density measure of other images near the first image x_{s1} will have a reduced density, thereby making them unlikely to be selected as the next cluster center. After the density measure of each cluster is revised, the image with the next highest density x_{s2} is selected and all the density measures are revised. This process is repeated until a sufficient number of cluster centers are obtained.

5.5.2 K-Means Clustering

K-means or C-means have been widely used in the area of image processing and pattern recognition [48],[49]. K-means algorithm partitions n vectors into c groups. After initial clusters have been decided, the k-means algorithm is applied to find the clusters center. When Euclidean distance is used as a measure of dissimilarity between image X_k in and the *i*th groups $G_i, i = 1,\ldots,c$ with a center of cluster c_i , the cost function which is a measure of the total distance between x_k and c_i is defined as follows:

$$J = \sum_{i=0}^{c-1} J_i = \sum_{i=0}^{c-1} \left(\sum_{x,k \in G_i} d(x_k - c_i) \right) \quad (5.5)$$

Where *c* is the number of clusters and d is the Euclidian distance between two vectors. The value J_i depends on the geometrical properties of G_i and the location of c_i .

We define the membership function u_{ij} as follows:

$$u_{ij} = \begin{cases} 1 \; if \left\| x_j - c_i \right\|^2 \le \left\| x_j - c_k \right\|^2 \; for \, k \ne j \\ 0 \;\; otherwise \end{cases} \tag{5.6}$$

x_i belongs to group i if the distance between x_i and c_i is closer than the distance between x_i and any other center.

c_i is updated by the mean of all vectors in a group as follows:

$$c_i = \frac{1}{|G_i|} \sum_{k, x_k \in G_i} x_k \tag{5.7}$$

This is repeated until the improvement over previous iteration is below a certain value. After all the images have been classified to one of the clusters, the image with minimum distance to the center of each cluster is chosen as a prototype to be used as a training sample. Since an image can belong to only one cluster, no image will be chosen more than once.

The number of features used for clustering is high; it would be desirable to reduce the number of features while preserving the properties of the clustering centers. For this purpose we use the Fisher discriminate or distance for each feature set to select the most effective features [49]. The Fisher distance is a measure of cluster separability. The mean and standard deviation of the feature value in each cluster are used to compute the Fisher distance

$$F_k = \sum_{i=0}^{c-1} \sum_{j=i+1}^{c-1} (\mu_{k,i} - \mu_{k,j})^2 / (\sigma_{k,i}^{\,2} + \sigma_{k,j}^{\,2}) \tag{5.8}$$

where k is feature number, $\mu_{k,i}$ is the mean of the feature k on cluster i and $\mu_{k,j}$ is the mean of the feature k on cluster j, $\sigma_{k,i}$ is the standard deviation of the feature k on cluster i and $\sigma_{k,j}$ is the standard deviation of the feature k on cluster j.

As can be seen from the (5.6), the Fisher distance increases as the difference between means of the feature values for the two clusters increase, or as the separability between the two clusters increase, In a similar manner as the

standard deviation of the feature in each cluster decrease, the Fisher distance increase. The features that generate the highest Fisher distance are considered effective.

From 21 features used for clustering, only seven features are chosen. The clustering is repeated and each time one of the less effective features is removed. If the clustering results remain unchanged another feature is removed. This is repeated until removing the one of the less effective feature changes the clustering results. It is found that the clustering result obtained using 21 features listed in Table 5.2 can be obtained using the seven features listed in table 5.1.

Table 5.2 The most effective features

No	Name
1	c_x
2	c_y
3	c_1
4	c_2
5	c_3
6	c_4
7	θ_6

Fisher distance is used to limit the number of inputs to the neuro-fuzzy system.

5.6 Template Matching

Template matching is referred to as the process of trying to find a template image in a target image. This is done by matching the template image to all (or many) of the possible places it could be located in the target image. A distance measure function (typically a simple Euclidean distance) or measure of normalized correlation is applied to the match to measure the similarity of the template and the image at that location. By sliding the template over the target image, the algorithm picks the location with smallest distance as the location of the template image moves over the target image as shown in Fig.5.13. This method requires high computation time, however, Many techniques have been developed to reduce this computation time [51]. One common technique for this is coarse-to-

fine template matching where a low-resolution version of the template is compared with low-resolution version of the image to find a candidate location of a good match and then higher resolution comparison is applied [52][53]. This kind of template matching requires less computation but it comes with a low precision.

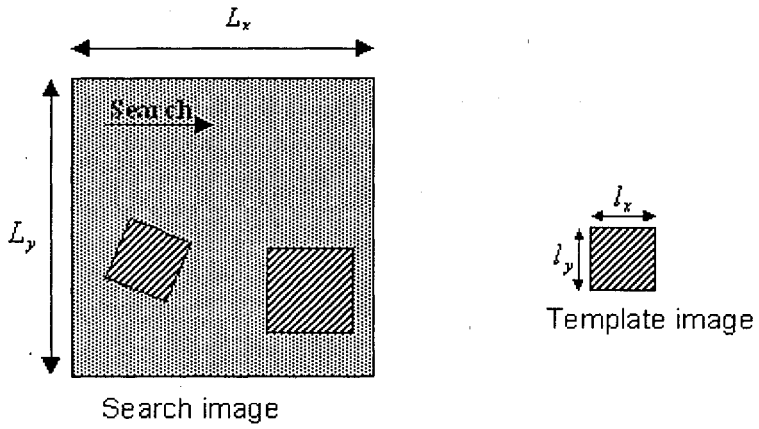

Figure 5.13 Template matching

There are some problems with conversional template matching methods. These methods cannot locate a rotated version of the template with high accuracy. To overcome this problem a dictionary of templates or different versions of the template are searched, which will drastically increase the computation cost. These methods are not robust against object deformation and environmental changes. Therefore, template matching techniques must be extended to overcome the above-mentioned problems. The new method must be fast and robust against deformations.

The parametric template method similar to that used by [54-57] achieves this through matching between a template space constructed from a set of templates. A linear combination model of multiple images is constructed where the linear parameters of the model are obtained from the results of feature-based corresponding-point matching.

Here we construct a parametric template space for each landmark. The template space is a linear combination of several templates as shown in Fig. 5.14.

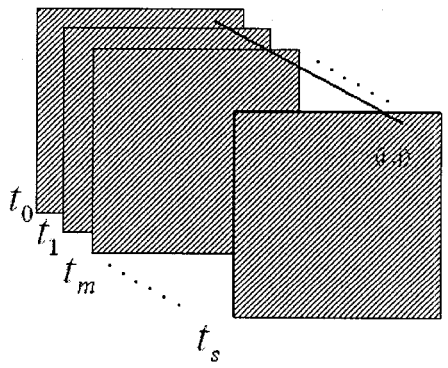

Figure 5.14 Construction of parameter space form a combination of templates of different rotations

The normalized template space is constructed as follows:

$$t_s = \frac{a_0 t_0 + a_1 t_1 + ... + a_m t_m}{\left\| a_0 t_0 + a_1 t_1 + ... + a_m t_m \right\|} \tag{5.9}$$

where $t_0, t_1, ..., t_m$ are the vertex templates and $a_0, a_1, ..., a_m$ are the linear parameters satisfying the following conditions:

1- $\{a_0, a_1, ..., a_m\}$ are all positive.

2- the sum of $a_0, a_1, ..., a_m$ equal to 1.0

This region thus forms a simplex having the base templates as its vertexes in the template space. Any template in the parametric template space is uniquely represented by a linear parameter a_i. Figure 5.15 shows the formulation of parametric template of rotated images for one landmark.

-5 -4 -3 -2 -1 0 1 2 3 4 5

Figure 5.15 Extracting a template with different rotations of one landmark

5.6.1 Parametric Template Matching

The normalized correlation between any two templates *g* and *t* of sizes $l_x \times l_y$ is defined as follows:

$$\rho = \frac{\sum_{(k,s)\in TW} (\Delta t) \times (\Delta g)}{\sqrt{\sum_{(k,s)\in TW} (\Delta t)^2} \sqrt{\sum_{(k,s)\in TW} (\Delta g)^2}} \tag{5.10}$$

$$\Delta t = t(k,s) - \bar{t} \tag{5.11}$$

$$\Delta g = g(k,s) - \bar{g} \tag{5.12}$$

$$TW = \{(k,s) \mid 1 \le k \le l_x, 1 \le s \le l_y\} \tag{5.13}$$

Where $t(k,s)$ and $g(k,s)$ are the gray level at location (k,s), and $\bar{t}$ and $\bar{g}$ are the mean values of the of images *t* and *g*.

Let $\hat{t}$ be the normalization of image t and $\hat{g}$ be the normalization of image g defined as follows:

$$\hat{t}(s,k) = \frac{\Delta t}{\sqrt{\sum_{(k,s)\in TW} (\Delta t)^2}} \tag{5.14}$$

$$\hat{g}(s,k) = \frac{\Delta g}{\sqrt{\sum_{(k,s)\in TW} (\Delta g)^2}} \tag{5.15}$$

Then the normalized correlation can be written as

$$\rho = \sum_{(k,s)\in TW} \hat{t}(s,k) \times \hat{g}(s,k) \tag{5.16}$$

The normalized image satisfies the following normalizing conditions:

$$\sum_{(k,s)\in TW} \hat{g}(s,k) = 0.0 \tag{5.17}$$

$$\sum_{(k,s)\in TW} \hat{g}(s,k) \times \hat{g}(s,k) = 1.0 \tag{5.18}$$

Extending the template matching between two templates to matching template with template space

$$\sum_{i=0}^{M} \omega_i \hat{t}_i \quad (0.0 \le \omega_i \le 1.0 \sum \omega_i = 1.0) \tag{5.19}$$

Normalizing the template space

$$*t_s = \frac{\sum_{i=0}^{M} \omega_i \hat{t}_i}{\left\| \sum_{i=0}^{M} \omega_i \hat{t}_i \right\|} \tag{5.20}$$

satisfies the normalization condition in (5.17) and (5.18)

A normalized correlation between the reference template g an a parametric template space is $(\hat{g}, *t_s)$. We are looking for the coefficients ω_i such that the correlation between the template space and the reference image is maximized. This is a constrained optimization problem solved using a Lagrange multiplier and the coefficients ω_i which can be obtained as follows:

$$\overline{\omega} = \frac{H^{-1}\vec{G}}{(\vec{n}, H^{-1}\vec{G})} \tag{5.21}$$

where

$$H = \begin{bmatrix} (\hat{t}_0, \hat{t}_0) & \ldots & (\hat{t}_0, \hat{t}_M) \\ . & . & . \\ . & . & . \\ . & . & . \\ (\hat{t}_0, \hat{t}_M) & . & (\hat{t}_M, \hat{t}_M) \end{bmatrix} \tag{5.22}$$

$$\vec{G} = \begin{bmatrix} (\hat{g}, \hat{t}_0) \\ . \\ . \\ . \\ (\hat{g}, \hat{t}_M) \end{bmatrix} \tag{5.23}$$

$$\vec{n} = \begin{bmatrix} 1 \\ . \\ . \\ . \\ 1 \end{bmatrix} \tag{5.24}$$

We can bind a features value e.g. a rotation or scaling into each vertex templates then formulate a template space that represents all of the deformations in the templates. In this way only one template is searched for while it carries all the properties of different deformed templates. In this matching algorithm, the optimal parameters ω which maximizes the correlation values and the features values are directly obtained from the normalized template space.

Chapter 6

Multi Layer Perceptron

6.1 Introduction

In this chapter we present automatic landmark detection utilizing an MLP neural network as a function approximator. The localization of landmarks is formulated as a function approximation problem. Locations of landmarks will not depend on the quality of the x-ray but on the size, rotation and shifts of the outer contour of the skull. The features extracted from images are used as inputs to the MLP. The MLP is trained by a number of labeled images and then used to predict the location of the landmarks on new images (not used for training). This is based on the knowledge obtained during the training, using the same features listed in Table 5.1.

Since it is not feasible to train the network to map all cases in its given domain, a small set of training data is to be determined such that the network would be generalized enough, and capable of learning from the training data, to perform on any case with acceptable tolerance. A method is used to cluster images in such a way that similar images are grouped together to form one exemplar of the human skull.

6.2 Selection of the Training Set

To come up with the prescribed training set, images had to be clustered into several groups based on their feature vector using K-means. In order for the K-means algorithm to converge, initial estimates of cluster centers are needed.

After extracting the features listed in Table 5.1, each x-ray image is represented by a vector of 21 elements. Because of the sensitivity of the clustering algorithms, all feature vectors are normalized beforehand so that parameters in the training set are uniformly distributed in the scaled range between 0 and 1.

The length of the image vector, which is the distance between the feature vector and the origin of the hypercube, is computed as follows:

$$\|L\| = \sqrt{\sum_{i=1}^{i=NF} (x_i)^2} \tag{6.1}$$

where $\{x_1, x_2, ..., x_{NF}\}$ is the feature vector and *NF* is the number of features. To provide initial estimates for K-means, several steps are followed. First the feature vectors are normalized and sorted in ascending order according to their length. These are then represented in a diagram similar to that of Fig.6.1. The horizontal axis represents the normalized length of the vectors. At each location a vertical line equal in length to the value of the normalized vector is placed to represent each image. The horizontal distance between 0 and 1 is divided into equal intervals of length 1/(number of samples). This will generate an estimate of the distribution of the images with respect to the length of their vectors. The number of samples was set to 55 images through empirical trials. After several trials in which intervals with zero number of vectors were discarded and any interval with relatively high number of vectors is divided into two, it was found that our set can be initially represented by 55 groups.

Starting with 189 images, of sizes ranging between 480×480 to 564×564 pixels, a smaller training set is formed using the K-means algorithm with initial estimates of centers obtained from the previous step.

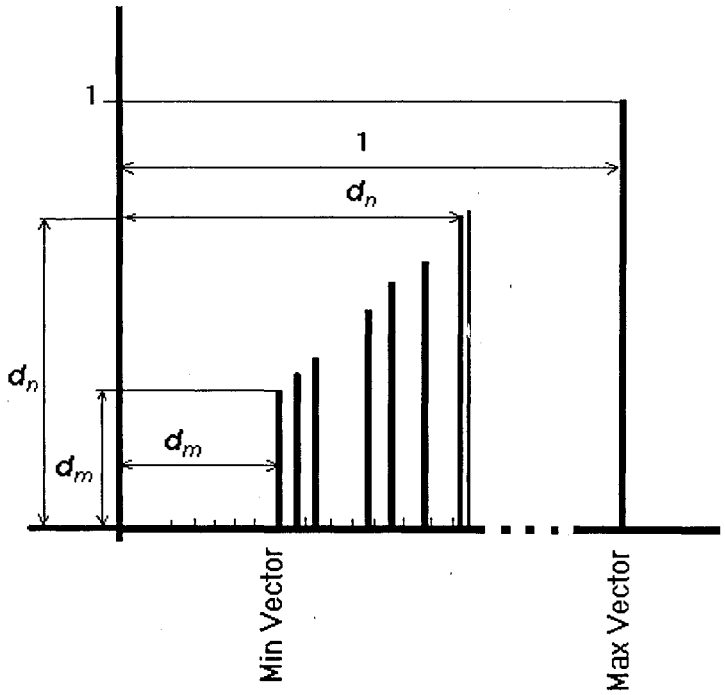

Figure 6.1 Initial clusters centers using normalized image lengths

After selecting the training set, all landmarks on the training set images are labeled by a human expert. The training algorithm uses feature vector as an input while the labeled locations are used as the desired output. From each group one image is selected as a prototype. The prototype image of each group is the image which is closest to the group center. By selecting a prototype image from each group as stated we have ensured that all variations in the images represented by any one cluster are included in the training set.

6.3 MLP as a Function Approximator

MLP's have been widely used in function approximation problems [58-60]. Estimating the location of landmarks can be formulated as a function approximation problem, for which MLP's are well suited due to their universal approximation property. Fig. 6.2 shows the block diagram for the proposed MLP.

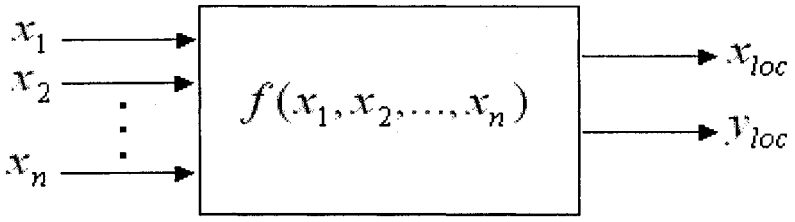

Figure 6.2 Block diagram of MLP

where $\{x_1, x_2, ..., x_n\}$ is the feature vector

The MLP is set up as a general approximator to `learn' the given input/output relations or the mapping function between the input and the output by updating the connection weights in the approximator according to the back propagation algorithm [61]. The MLP is presented with the different training item until it learns the mapping of the inputs to the proper output. The learning process is terminated when some error criterion are met. Based on the Stone-Weierstrass theorem, Cotter [62] suggested that a two-layer feedforward network could be used to approximate any continuous function by mapping variables from $\Re^n$ to $\Re^m$. The task of approximating an unknown function from input-output pairs can be formulated as follows:

The input-output pairs are given as, $T = \{(X_p, Y_p); p = 1,...,P\}$ where P is the number of training pairs. The input vector $X_p = \{x_{p1},...,x_{pn}\}^T$ and the output vector $Y_p = \{y_{p1},...,y_{pn}\}^T$ are related by unknown function f such that $Y_p = f(X_p) + e_p$. The e_p is an error vector. The task of the MLP is to find an estimator f' for f such that some metric for the error is minimized.

In this research f is the horizontal location of the landmark as obtained by a human expert. We are seeking f' such that the predicted values obtained from f' are close to the values of f. In this research f is known only at discrete values and we wish to derive an approximation that is valid at points not necessarily found in the training set but that are expected to occur in the test set

or in other images outside the test or training sets. Since f is not continuous, the discrete values are made small enough so that the MLP can interpolate between values. Fig. 6.3 shows a one-dimensional view of the approximation. There are 21 axis representing the image features. The vertical axis's represents the x or y locations of the landmark. Since the output of the MLP is between 0 and 1, all locations of the landmarks are previously normalized.

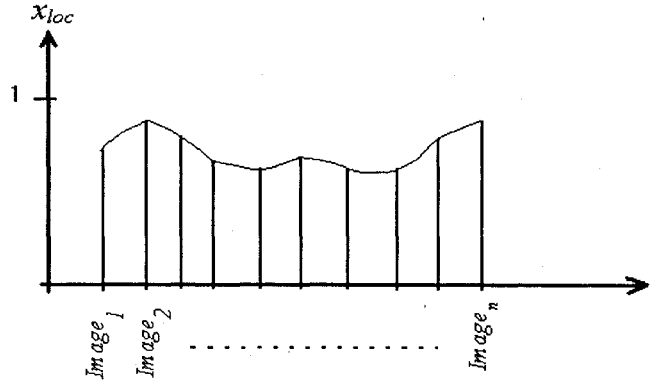

Figure. 6.3. Horizontal location represented as a function of the projection of one feature in one feature axis

The back propagation algorithm provides a means of training the MLP to perform supervised learning tasks by adjusting the connection weight in such a way that the difference between the input and the desired output is minimized. After the MLP is trained and the connection weights are obtained, the MLP is tested on target image (images that are not used in training). In the testing stage the value of the output is scaled back by multiplying by the width of the image.

The proposed MLP is a three layer network. An input layer, a hidden layer and an output layer as shown in Fig 6.4. MLP approximates function by ensembles of simpler function as follows.

Let $f_x(\mathrm{X})$ and $f_y(\mathrm{X})$ be the functions for approximating the horizontal and vertical coordinates of a landmark, where $X=[x_1, x_2, ..., x_n]^T$ is the feature vector, the goal of the function approximation is to describe $f_x(\mathrm{X})$ and $f_y(\mathrm{X})$ by combinations of simpler functions $\varphi(\mathrm{X})$ and $\lambda(\mathrm{X})$:

$$\hat{f}_x(\mathrm{X},\mathrm{W}) = \sum_{i=1}^{N} w_i \phi_i(\mathrm{X}) \tag{6.2}$$

$$\hat{f}_y(\mathrm{X},\mathrm{W}) = \sum_{i=1}^{N} w_i \lambda_i(\mathrm{X}) \tag{6.3}$$

where *N* is the number of ensembles and w_i are real-valued entries representing the connection weights coefficient vector $W = \{w_1, w_2, ..., w_m\}$ such that

$$\left| f_x(X) - \hat{f}_x(X,W) \right| < \varepsilon_x \tag{6.4}$$

$$\left| f_y(X) - \hat{f}_y(X,W) \right| < \varepsilon_y \tag{6.5}$$

ε_x and ε_y are minimizing a least mean squared sense to obtain an approximation for the desired output vector.

The basis function can be wavelets, sinc function, or polynomial. Since $f_x(X)$ and $f_y(X)$ are both nonlinear functions, there is no natural choice of the proper number of basis functions (for our application we used the sinc function). If too many terms are used, we will be faced with an over-fitting problem and an under-fitting problem if the number is too small. The best approximation is obtained with a minimal number of basis functions which we can obtain by experimentation. The basis will change with input data, which means the weights in the input layer changes the orientation of the basis while the weights in the output layer find the proper amplitude of the units [62-63].

After the MLP is trained using the training set, it is used to approximate the location of the landmarks based on the knowledge obtained during the training phase. The algorithm was tested on 134 x-ray images which were not used for training.

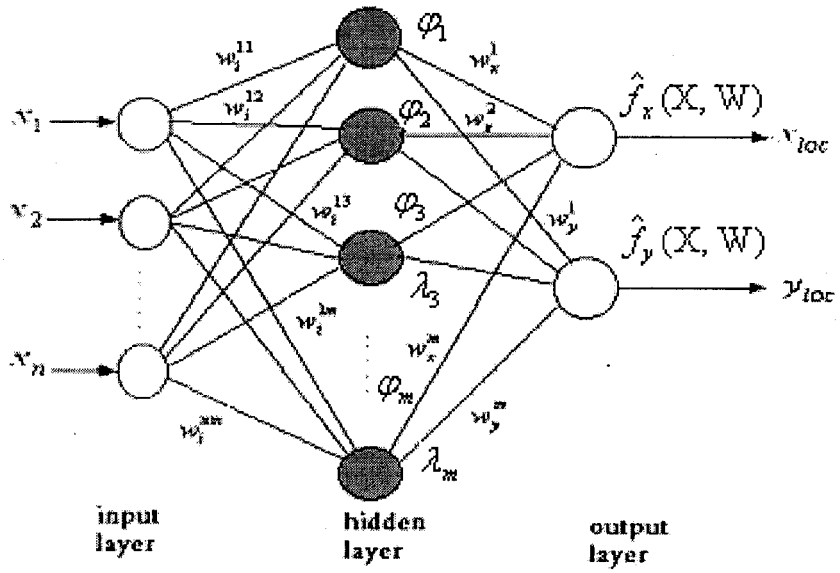

Figure 6.4. Three layers MLP

Results obtained by the algorithm are compared to those obtained by a human expert. If the difference between the approximated result (obtained form the MLP) and the expected result (obtained from human expert) is less than or equal ± 2mm, then the approximation is considered successful and acceptable otherwise it is considered a failure.

Chapter 7

Experimental Results

7.1 Introduction

In the previous chapters we have provided the recourses for designing pattern recognition systems for cephalometric evaluation. The basis for the proposed systems is neuro-fuzzy and neural networks. The proposed systems use image processing techniques to extract a feature vector from an x-ray image which is then used as an input to the system to predict the locations of the landmarks. The first system uses a neuro-fuzzy network to predict a smaller search window for each landmark. After the search windows are found, a template-matching algorithm based on template space is utilized to pinpoint the exact location of the landmark. The second system uses a Multi-Layered Perceptron (MLP) network as a function approximator to directly predict the locations of landmarks. In this chapter, we provide the results obtained using the two systems. The two algorithms were tested on a set of 189 x-ray images of both gender ranging form 9 to 39 years of age. The sizes of the x-ray images range form 127.8x127.8 mm to 156.4x156.4 mm with resolution from 460x460 to 564x564 pixels and 256 gray-levels. With this resolution, position accuracies of about 0.28 mm are at best possible.

This resolution is more than adequate considering that the accepted normal range of most measurements is about ±2 mm.

The test results obtained are compared against manually located landmark positions. The distance in mm between the computer located position and the positions extracted manually by the expert is recorded as location error of the algorithms. This error should be within the range of ±2 mm for the position of the landmark obtained by the algorithm to be a considered a success. The x-ray images are divided into two sets; the first set is used for training and the second set is use for testing the algorithms. The test set does not include images used for training. We finish the chapter by comparing the results obtained here with results obtained by other researchers.

7.2 Selection of Test Landmarks

A total of 20 assorted landmarks was selected from the Atlas of Cephalometry [1] for experiments. The landmarks were of varying types and should provide an adequate test base for assessing the proposed algorithms. Fig.7.1 shows the locations of the landmarks of the x-ray and Table 7.1 lists the selected landmarks

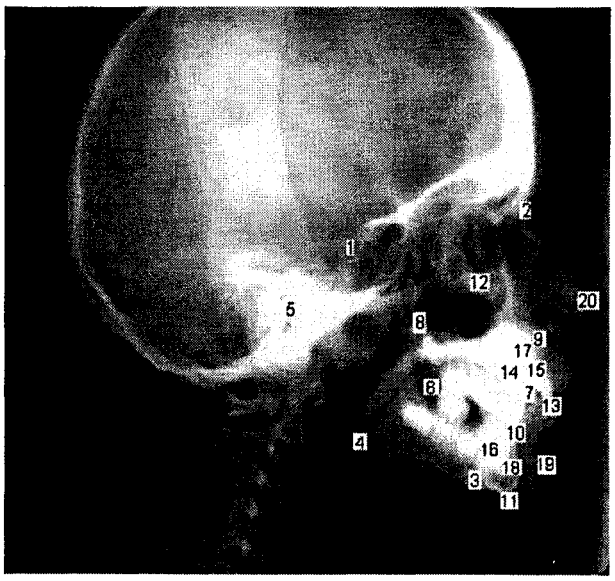

Figure 7.1 The test landmarks use in the experiments

Table 7.1 Defined Test Landmarks

No.	Landmark	Abbreviation
1	Sella	S
2	Nasion	Na
3	Menton	Me
4	Gonion	Go
5	Porion	Po
6	Posterior Point for occlusal plane	PPOcc
7	Anterior point for occlusal plane	APOcc
8	Posterior nasal spine	PNS
9	Anterior nasal spine	ANS
10	Pogonion	POG
11	Gnathion	Gn
12	Orbitale	Or
13	Incisor superious	Is 1
14	Incisor inferious	Is
15	Apicale $\bar{1}$	AP $\bar{1}$
16	Apicale $\underline{1}$	AP $\underline{1}$
17	Point A, subspinale	A
18	Point B, superamentale	B
19	Soft tissue pogonion	SoftPog
20	Tip of the Nose	Nose

7.3 Features Extraction

For easy and fast access of the x-ray images and landmarks, a database containing two tables was constructed. One table contains the x-ray image, age, sex, address, and all information of the patient and the second contains landmarks and their definitions with the ability or delete a landmark as shown in Fig. 7.2.

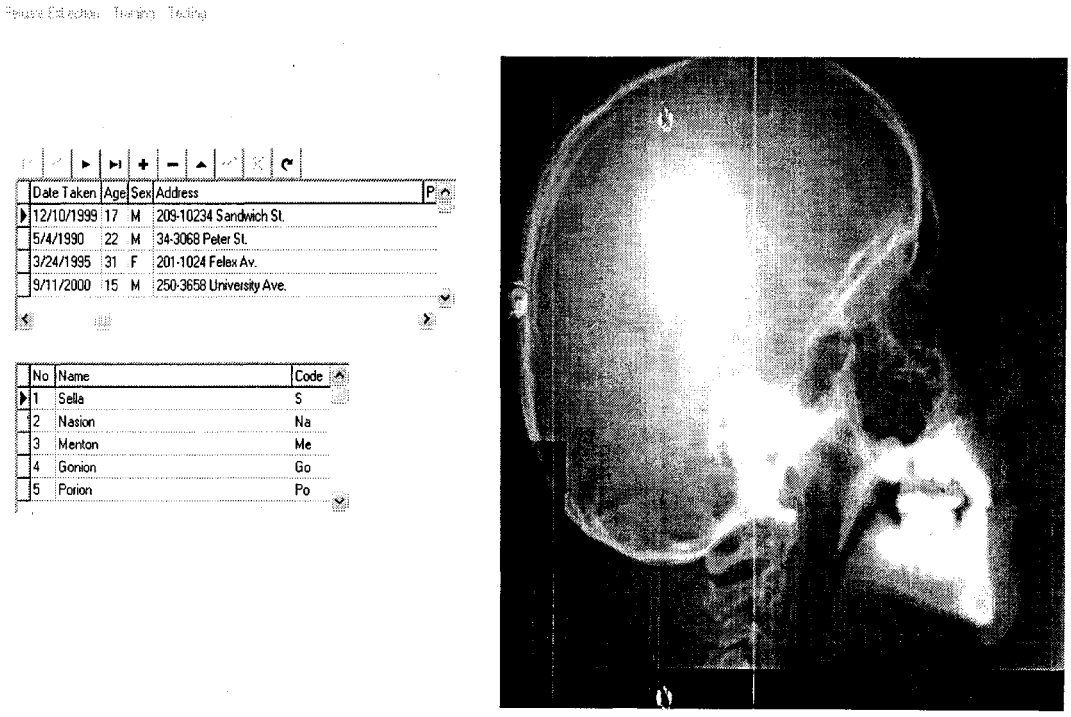

Figure 7.2 Two tables: one for patient information and the other for landmarks

Image enhancement, median filtering, edge detection and features extraction of all image is done all at once. Due to image fading in some x-rays, point one is not always detected as shown in Fig. 7.3. In this case manual intervention is required and the location of point P1 is determined manually. This situation was encountered in about 8% of the images. All features are extracted form all images, normalized, and stored in a file referring each feature vector by the image number in the database.

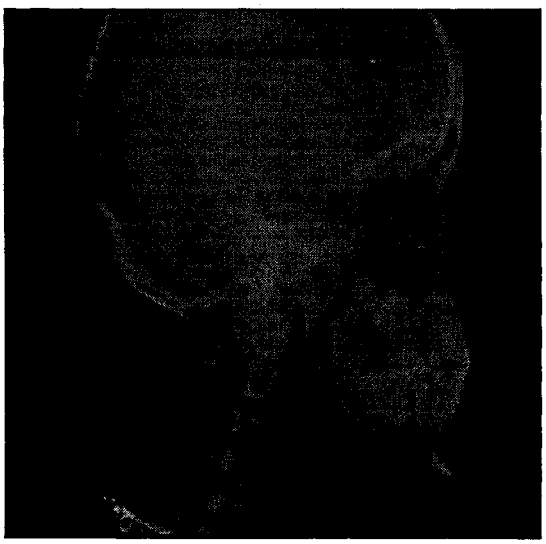

Figure 7.3 Original image

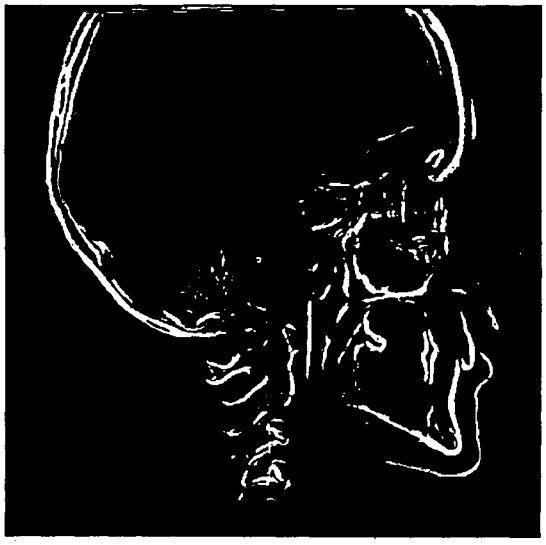

Figure 7.4 Manual intervention is needed to detect point P1

7.4 Experiment 1

Neuro-Fuzzy Network with Template Matching

After applying the Fisher distance, outlined in section 5.4.2, the number of input variables is reduced to seven variables given in table 5.3. Data is clustered using k-means after initial centers are determined using subtractive clustering. One image is selected for each cluster based on its distance from the center of the cluster. The training set consists of images selected from each cluster. From the training set, fuzzy sets representing each feature are constructed based on the variation of the feature on the training set using the Generalized Bell Membership Function given by (4.5). By training, the variables *a* and *b* can be obtained.

Example of fuzzy set representing the feature c_1, which is a measure of the distance between the center of gravity and point P1, is shown in Fig. 7.5. The center of the first set is the minimum value of the feature and the center of the second set is maximum values of the feature.

The first step is to test if all landmarks are within their search windows. Table 7.2 summarizes the results obtained for landmarks falling within a window of 40x40pixel and 50x50 pixels. Images that have landmarks outside 50x50 search windows are not included in the test set.

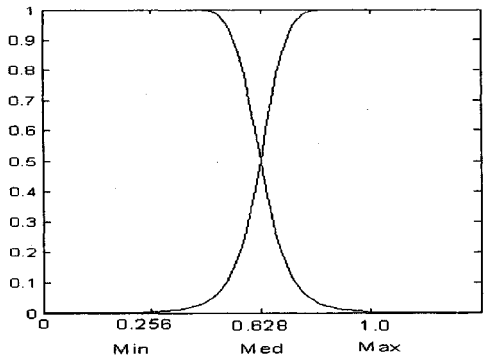

Figure 7.5 Fuzzy sets representing of feature c_1

Table 7.2 Percentages of landmarks within the search windows

No.	Landmark	Within 40x40	Within 50x50
1	S	94.8%	98%
2	Na	97.4%	100%
3	Me	100%	100%
4	Go	83.1%	93%
5	Po	94.1%	97.4%
6	PPOcc	96.1%	98.7%
7	APOcc	98.1%	98.7%
8	PNS	96.75%	99.4%
9	ANS	98.05%	98.7%
10	POG	100%	100%
11	Gn	100%	100%
12	Or	98.7%	99.3%
13	Is 1	98.05%	98.7%
14	Is	98.05%	98.7%
15	AP $\bar{1}$	96.1%	99.3%
16	AP 1	98.05%	99.3%
17	A	93.5%	98.0%
18	B	100%	100%
19	SoftPog	99.35%	100%
20	Nose	100%	100%

For the template space construction, templates of size 20x20 are chosen from a better than average quality image. Because of the variations in sizes of the shape representing the landmarks, some landmarks need more than one template to represent the shapes as shown in Fig.7.2. It is easer to locate a complex shape, if it is broken down into several shapes [13]. The template space is constructed as outlined in Section 5.5.1.

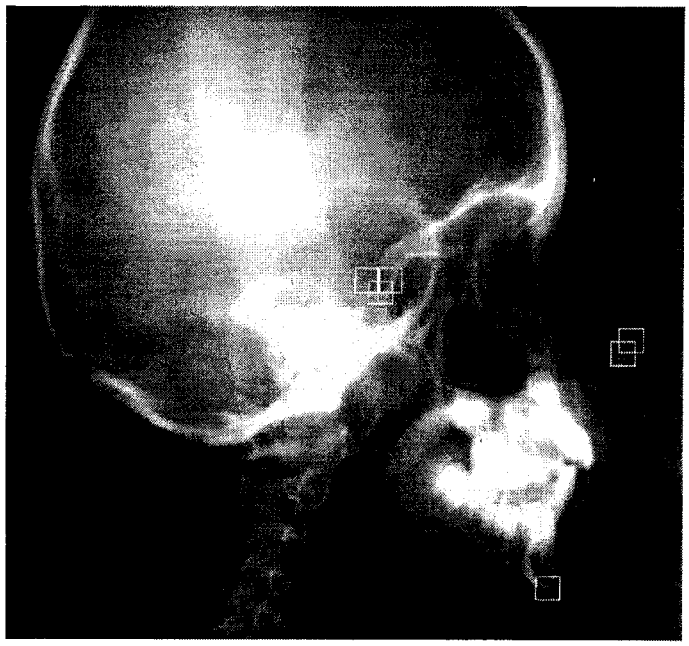

Figure 7.6. Extraction of template from different landmarks

After extracting templates, the template space is constructed for each landmark. The search window is limited to 50x50 pixels for all landmarks. The location of the landmark is considered successful if the difference between the position of the landmark obtained by the algorithm and the position labeled by the operator is less than ±2 mm or less than ±7 pixels.

The results have been summarized in Table. 7.3. This table lists the percentage of correct location of the landmarks in the test set.

Table 7.3 Percentage of landmarks located correctly

No.	Landmark	Percentage of correct location Within ±2 mm
1	S	77%
2	Na	95%
3	Me	100%
4	Go	87%
5	Po	94%
6	PPOcc	73%
7	APOcc	76%
8	PNS	100%
9	ANS	93%
10	POG	100%
11	Gn	100%
12	Or	77%
13	Is $\overline{1}$	79%
14	Is $\bar{1}$	95%
15	AP $\bar{1}$	100%
16	AP $\overline{1}$	100%
17	A	88%
18	B	86%
19	SoftPog	100%
20	Nose	100%
Total		91%

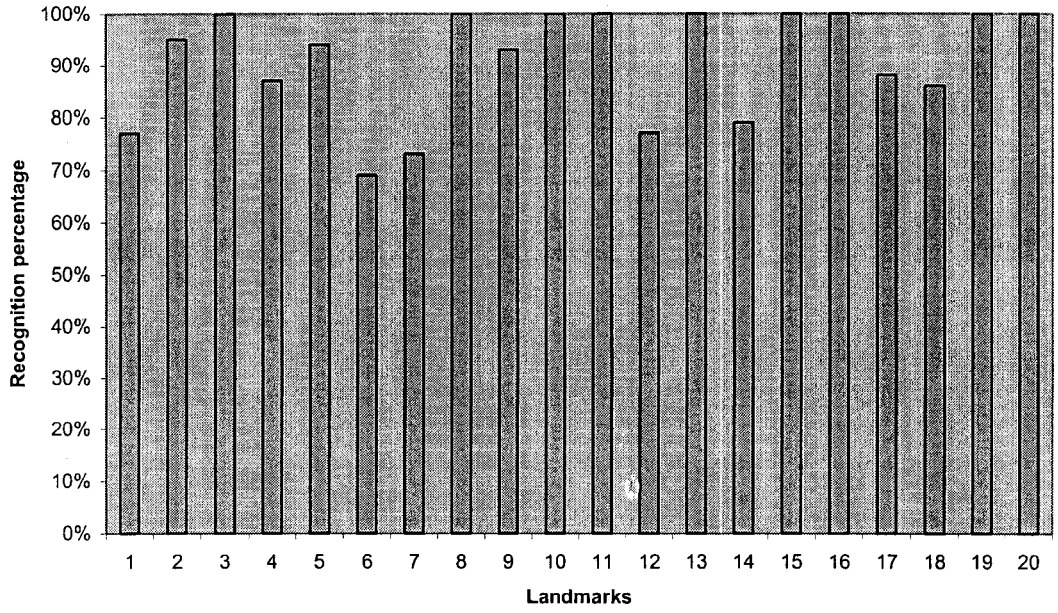

Figure 7.7 Recognition rates of the landmarks

Discussion: Form Fig. 7.7 we note that most of the landmarks can be accurately located with a rate above 86%. Some of the landmarks cannot be located accurately due to one or more of the following reasons:

1-The shape recognition algorithm cannot recognize the shape of a landmark having high morphological variations such as landmark incisor superiors (13) and landmark Sella which have different shapes in different images as shown in Fig 7.8 and Fig 7.9.

2- The shape sought for is missing. This occurs when there is a missing tooth or when there is a cavity filling instead of the tooth. This is shown in the Fig. 7.10 for the landmark APOcc (7) which is defined as the most distant point of contact on the most posterior molar.

It will be difficult for any shape-matching algorithm to resolve such a problem. Instead of using shape matching algorithm, it will be easer to overcome the mentioned problems by using an ANN to predict the location directly by formulating the problem as a function approximation.

Figure 7.8. Different shapes of landmark Incisor superius.

Figure 7.9 Different shapes of landmarks Sella

Figure 7.10 Different cases of landmark PPOcc

7.5 Experiment 2

Multi Layer Perceptron (MLP)

In this experiment we use MLP as outlined in Chapter 6. In this experiment the MLP is trained using larger number of exemplars. All images are clustered into 55 groups. From each group one image, which is closest to the center of groups, is chosen as a representative of that group. After being trained the MLP will be able to predict the location of landmarks based of the knowledge obtained during the learning stage.

Fig.7.11 shows the location of landmarks on the lateral x-ray of the skull using the proposed technique. Table 7.4 shows the percentage of successfully located landmarks and a comparison with results obtained using the neuro-fuzzy approach. Table 7.5 shows a comparison of results obtained with those obtained by other researchers in [13][14].

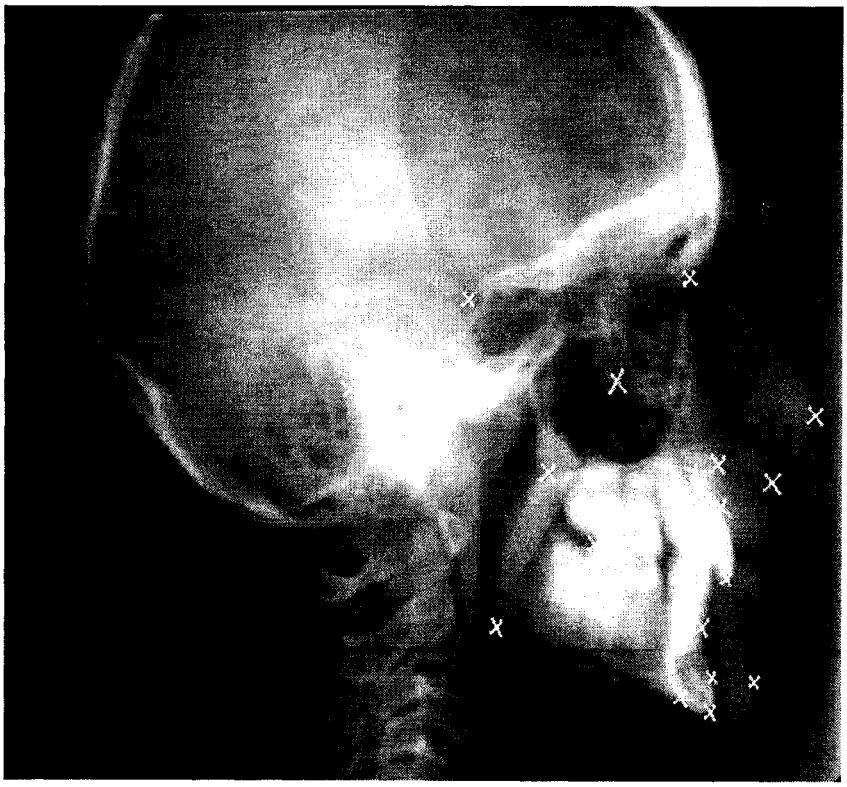

Figure 7.11. Localization of landmarks on the skull x-ray

Table 7.4 Comparison of results obtained using the two approaches

No.	Landmark	Recognition Rate using NFNN	Recognition Rate using MLP	Difference between the two systems
1	S	77%	85.00%	8%
2	Na	95%	91.00%	-4%
3	Me	100%	92.00%	-8%
4	Go	87%	85.00%	-2%
5	Po	94%	97.00%	3%
6	PPOcc	73%	87.00%	14%
7	APOcc	76%	68.00%	-8%
8	PNS	100%	85.00%	-15%
9	ANS	93%	90.00%	-3%
10	POG	100%	100.00%	0%
11	Gn	100%	97.00%	-3%
12	Or	77%	74.00%	-3%
13	Is l	79%	88.00%	9%
14	Is $\bar{\mathrm{l}}$	95%	100.00%	5%
15	AP $\bar{\mathrm{l}}$	100%	97.00%	-3%
16	AP l	100%	95.00%	-5%
17	A	88%	91.00%	3%
18	B	86%	85.00%	-1%
19	SoftPog	100%	97.00%	-3%
20	Nose	100%	96.00%	-4%
Average		91%	90%	-1%

Table 7.5 Comparison of results obtained using NFNN with that reported by [13]- and [14]

No.	Landmark	Cardillo [13] 1994	Grau [14] 2001	Using NFNN 2003	Improvements over [13]	Improvements over[14]
1	S	53%	65%	77%	24%	12%
2	N	83%	85%	95%	12%	10%
3	Me	78%	100%	100%	22%	0%
4	Go	61%	85%	87%	26%	2%
5	Ar	89%		94%	5%	
6	PPOcc	71%		73%	2%	
7	APOcc	48%		76%	28%	
8	ANS	68%	90%	100%	32%	10%
9	PNS	71%	80%	93%	22%	13%
10	Pog	97%	95%	100%	3%	5%
11	Gn	100%	90%	100%	0%	10%
12	Or	40%	65%	77%	37%	12%
13	Is	76%	100%	79%	3%	-21%
14	Li	64%	90%	95%	31%	5%
15	Ap	89%	100%	100%	11%	0%
16	AP 1	79%	100%	100%	21%	0%
17	Point A	77%	95%	88%	11%	-7%
18	Point B	71%		86%	15%	
19	SoftPog	91%		100%	9%	
20	Nose	94%		100%	6%	
	Average	**75%**	**89%**	**91%**	**16%**	**2%**
	Test Set	**20**	**20**	**154**		

Table 7.6 Comparison of results obtained using MLP with those reported by [13] and [14]

No.	Landmark	Cardillo [13] 1994	Grau [14] 2001	Using NFNN 2003	Improvements over [13]	Improvements Over[14]
1	S	53%	65%	85.00%	32%	20%
2	N	83%	85%	91.00%	8%	6%
3	Me	78%	100%	92.00%	14%	-8%
4	Go	61%	85%	85.00%	24%	0%
5	Ar	89%		97.00%	8%	
6	PPOcc	71%		87.00%	16%	
7	APOcc	48%		68.00%	20%	
8	ANS	68%	90%	85.00%	17%	-5%
9	PNS	71%	80%	90.00%	19%	10%
10	Pog	97%	95%	100.00%	3%	5%
11	Gn	100%	90%	97.00%	-3%	7%
12	Or	40%	65%	74.00%	34%	9%
13	Is	76%	100%	88.00%	12%	-12%
14	Li	64%	90%	100.00%	36%	10%
15	Ap	89%	100%	97.00%	8%	-3%
16	AP 1	79%	100%	95.00%	16%	-5%
17	Point A	77%	95%	91.00%	14%	-4%
18	Point B	71%		85.00%	14%	
19	SoftPog	91%		97.00%	6%	
20	Nose	94%		96.00%	2%	
	Average	**75%**	**89%**	**90%**	**15%**	**1%**
	Test Set	**20**	**20**	**134**		

7.6 Conclusions

From table 7.4 we notice that there are improvements in the recognition rate of some landmarks using MLP over the neuro-fuzzy approach. This can be clearly seen in the recognition rate of landmarks Sella (+8%), Ppocc (+14%), and Is$_1$ (+5%). While the neuro-fuzzy approach outperforms the MLP approach in most of the other landmarks, the average recognition rate of the overall performance is almost the same. The MLP approach has the advantages of faster location of landmarks while it suffers from the length of the time required for training. The neuro fuzzy approach has faster training time but suffers from a lengthy period required for template matching in search windows.

Comparing the results obtained by the two systems shown improvement over the previously reported results obtained [13] and [14].

Chapter 8

Conclusions and Future Work

8.1 Summary

The main objective of this research was to develop an efficient algorithm to overcome the tedious work of manually locating craniofacial landmarks on cephalograms. This process usually requires experienced orthodontists about 20 minutes to perform.

In this thesis, several pre-processing techniques are used to enhance x-ray images and extract outer edges of the skull. Four points are automatically located on the outer edges of the skull. From the selected four points more features representing distances, angles between lines connecting the four points and center of gravity of the polygon shape of the four points are extracted. A feature vector of 21 elements representing rotations, shifts in the image of the skull, and distances between certain points in the outer contour of the image is obtained for each x-ray.

The extracted features are used to cluster images into several groups based on the similarity of their extracted features. After clusters are created, one image is selected from each cluster as a representative of that cluster. Using this approach we ensure that all variations of

images are present in the training set. After obtaining the training set, the features vector is used as the input to the neuro-fuzzy system. Since the number of extracted features is high, (21), for use in neuro-fuzzy system, the Fisher discriminant distance was used to reduce the number of elements in the feature vector so that only the most effective features are used while the clustering results are preserved. After reduction of the dimension of feature vectors to seven, we use a neuro-fuzzy system for reducing the search window for each landmark by training the system based on fuzzy rules extracted for the feature values of images of the training set. After minimizing the search window for each landmark, parametric template matching is used to pin point the exact location of the landmark. This algorithm had some drawback due to the nature of template matching techniques. It produced low recognition rate for some landmarks because of the large variations of landmark shapes and other anomalies such as missing tooth or cavity filling. To overcome the difficulties outlined in the previous algorithm, we used MLP as a function approximator to directly predict the expected location of the landmark. The MLP is trained using a set of labeled images then used the knowledge obtained during training to find the location of the landmark in the test set. Although MLP's are faster in locating landmarks, their training required longer time than the neuro-fuzzy system. Function approximation results using MLP required larger training sets but produced better results in locating landmarks that have high irregularities. Results obtained by the systems were compared with results obtained by other researchers and it was shown that the two approaches presented outperformed previous work found in the open literature.

8.2 Contributions

This thesis reports the following contributions:

1- A new feature vector, which is capable of measuring the morphology of the skull, which was implemented and used to classify images into several groups based on similarities between their feature vectors.

2- A new algorithm based on the use of neuro-fuzzy networks to drive a smaller expectation window for the landmark based on the use of if-then fuzzy inference. The dimension of the feature vector was reduced to limit the number of fuzzy rules that can be handled by the proposed system. The landmarks were pin pointed using a search template constructed from several deformations of the landmark.

3- A new algorithm based on the use of MLP as a function approximator to directly locate the landmark was implemented. The algorithm uses feature vectors as inputs and the horizontal and vertical locations of the landmark as its output. After being trained using a set of labeled images, the MLP will be able to interpolate between labeled locations and predict a location of the landmark based on the size, rotation and shifts of the images measured by the feature vector. The main advantage of this algorithm is that it is independent of the quality of the x-ray image.

4- The two algorithms were tested on a large test set (155 images for neuro-fuzzy system and 134 for the MLP). Recognition rates obtained from the two algorithms are over 90%. Comparing with results obtained by other researches, the proposed systems outperform all previous algorithms reported.

8.3 Conclusions

Experimental results showed that on average 90%-91% of the landmarks would be located within ±2mm of the expected position obtained by the human expert. Any error in locating landmarks below ±2mm will not effect the cephalometric evaluation and thus considered satisfactory. On the other hand, errors exceeding this amount, influence the measurements beyond the acceptable amount. The performance of the shape recognition algorithm is highly influenced by the size reduction of the search window. Recognition of 91% while operating on such a diverse shapes and inconsistent x-ray image quality is considered very good.

Due to the high variability of images, these results are very difficult to improve without an improvement of the image captured. In this context, direct digital radiography would eliminate some of the noise, and alteration added to digitized x-ray images. A lot of images used in these experiment suffer from lose of details. This loss of details can be due to bad exposure or anomalies (e.g., missing tooth, filled cavity, etc.).

In conclusion a number of points can be made: This work has shown that it is possible to automate the location of landmarks in cephalograms. The proposed systems are capable of comparable performance with the best solutions, and display the adaptability required to handle low quality images. Quantitative comparison with previous works showed superiority of the proposed algorithms considering the fact that the proposed system is tested using a larger number of test sets and used images that were digitized using a frame digitizer.

8.4 Suggestions for future work

Improvement of x-ray imaging, both in terms of geometric positioning of the subject and image quality itself would be valuable in assisting all techniques of measurements. Without improving the quality of image captured, it would be very hard to improve these results using the same image quality. Further work is necessary to achieve a better performance system. The following recommendations come from experimental observation and recognize the limitations imposed by this implementation of the algorithms.

1- Obtaining a better training set might take years of gathering x-ray images. A limiting factor in this is, however, the need to keep x-ray exposure as low as possible. It would help to improve the performance of the systems if different x-ray images could be obtained for the different categories such as age, gender, and race. In this case one model for each category can be constructed.

2- Fusion of more than one algorithm together will utilize the strength of each algorithm in locating a certain landmark. An early test for missing teeth, on

the x-ray, would be useful in determining which algorithm will be used. Cavities could be detected because of their brightness; "educated guesses" could be made to decide which algorithm should be used.

3- Improvement of the performance of the algorithms can be obtained if larger set of images is used for training. In this case function approximation will be more accurate and interpolation between points will be smaller. This, however, will not always produce better results since it might cause what is known as "function overfitting". An optimal training set can be obtained experimentally.

4- Other function approximator such as Radial Base Functions (RBF) might produce better function approximation.

5- If an x-ray were to be considered as one in series, it would be possible to use results obtained from x-ray taken 6 months or a year earlier as the initial estimates for the search windows.

6- To improve the shape recognition results, it would be useful to use more than one shape model.

7- It is common (though not images used in this research) to have a variety of other structures visible in cephalogram images such as rulers, radiation-limiting wedges and head-positioning devices. These structures will inevitably interfere with the successful location of points P1 and P3. Additionally, many cephalostat systems exclude the back of the head, making location of P2 andP4 impossible. Without these points the system is unable to proceed. To address this issue the, different algorithms might be required for landmarking cephalograms from a variety of sources.

References

[1] Rakosi T., *An Atlas and Manual of Cephalometric Radiology*, London: Wolfe Medical Publications,1982 .

[2] Lévy-Mandel A., Venetsanopoulos A, Tsotsos J. "Knowledge-based Landmarking of Cephalograms," *Computer and Biomedical Research*, vol.19, pp: 282-309, 1986.

[3] Yan CK, Venetsanopoulos A., Filleray E. "An Expert System for Landmarking of Cephalograms," *Proceedings of the Sixth International Workshop on Expert Systems and Applications*, pp: 337-356, 1986.

[4] Parthasaraty S., Nugent S., Gregson P., "Automatic Landmarking of Cephalograms," *Computer and Biomedical research,* vol. 22, pp: 248-269,1989.

[5] Tong W., Nugent S., Gregson P., Jensen G., Fay D. "Landmarking of cephalograms using a microcomputer system," *Computers and Biomedical Research,* vol 23, pp: 358-379, 1990.

[6] Davis D. N., Taylor C. J. "A blackboard Architecture for Automating Cephalometric Analysis," *Journal of Medical Informatics,* vol. 16, pp: 137-149, 1991.

[7] Forsyth D. B., Davis D. N. "Assessment of an automated cephalometric analysis system," *European Journal of Orthodontics vol.18,* pp: 471-478, 1996

[8] Contereras-Vidal J. L., Garza-Garza J., "Knowledge-Based System for Image Processing and Interpolation of Cephalograms," *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, Ottawa, On. Sept. 4-6 1990, pp: 75.1.1-75.1.4

[9] Jackson P. H., Dickson G. C., Birnie D. G., "Digital Image processing for Cephalometric Radiographic: A preliminary Report," *British Journal of Orthodontists*, vol. 12 pp: 122-132, 1985.

[10] Cohen A. M., Linney A. D., "A Low Cost System for Computer-Based Cephalometric analysis," *British Journal of Orthodontics*, vol. 13, pp: 105-108, 1986.

[11] Cohen A. M., Linney A. D. , "A preliminary study of Computer Recognition and Identification of Skeletal Landmarks as a new Method of Cephalometric Analysis," *British Journal of Orthodontics,* vol. 11, pp: 143-154. 1984.

[12] Davis D., "Knowledge-Based Cephalometric analysis: A Comparison with Clinicians using interactive Computer methods," *Computer biomedical Research,* vol. 27, pp: 10-28, 1994.

[13] Cardillo J., Sid-Ahmed M. A., "An Image Processing System for the Automatic Extraction of Craniofacial Landmarks," *IEEE Trans. In Medical Imaging*, vol. 13, No.2, pp: 275-289, June 1994,

[14] Grau V., Juan M. C., Monserrat C., Knoll C., " Automatic Localization of Cephalometric Landmarks, " *Journal of biomedical information,* vol.34 pp: 146-156, 2001.

[15] Desvignes M., Romaniuk B. M., Demoment R., Revenu M., Deshayes M.J. "Computer assisted landmarking of cephalometric radiographs," Image Analysis and Interpretation, 2000. Proceedings. 4th IEEE Southwest Symposium2000, pp: 296-300, 2000.

[16] Hutton T. J., Cunningham S., Hammond P., "An evaluation of active shape models for the automatic identification of cephalometric landmarks," *The European Journal of Orthodontics*, vol. 22, Issue 5, pp: 499-508 October 2000.

[17] Rudolph D. J., Sinclair P. M., Coggins J. M., "Automatic computerized radiographic identification of cephalometric landmarks," *American Journal of Orthodontics and Dentofacial Orthopedics,* vol.113, pp: 173-179,1998.

[18] Uchino E., Yamakawa K., "High speed Fuzzy Learning Machine with Guarantee of Global Minimum and Its Application to Chaotic System Identification and Medical Image Processing" *Seventh International Conference Proceedings on Tools with Artificial Intelligence*, pp: 242 - 249, Nov. 1995.

[19] Sanei S., Sanaei P., Zahabsaniei P., "Cephalogram analysis applying template matching and fuzzy logic," *Image and Vision Computing* , vol.18, pp:39–48,1999.

[20] Chen Y. T., Cheng K. S., Liu J. K. "Improving Cephalogram Analysis through Feature Sub-image Extraction," *IEEE Engineering in Medicine and Biology*, vol. 18, pp: 25-31,1999.

[21] Andrew I., Ciesielski V., Mamutil J., Sabu J., "Landmark Detection for Cephalometric Radiology Images using Pulse Coupled Neural Networks," International conference in Computing in Communications pp: 391-396, June 2002.

[22] Broadbent HB., "A new X-ray technique and its application to Orthodontia," *Angle Orthod*, pp: 1:45, 1931.

[23] Brodie Ag., "On the Growth Pattern of the Human Head from the Third Month to The Eighth Year of Life,". *Am J Anat.,* vol.34, pp:812-840 1948.

[24] Downs Wb., "Variation on Facial Relationships: Their significance in treatment and Prognosis," *Am. J. Orthod,* vol. 34, pp: 812-840, 1941.

[25] McCulloch W., Pitts W., " A logical Calculus of the Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, vol.5, pp: 115 – 133, 1943.

[26] Sid-Ahmed M. A., *Image Processing Theory, Algorithms and Applications*, McGraw Hill, New York, 1994.

[27] Principe J., Euliano N., Lefebvre W., Neural and Adaptive Systems, New York: Wiley & Sons, pp.238-242, 2000.

[28] Zadeh L. A. , "Fuzzy Sets," *Information & Control*, vol. 8, pp: 338-353, 1965

[29] Zadeh L. A. , "Fuzzy algorithms," *Information & Control.*, vol. 12, pp: 94-102, 1986.

[30] Zadeh L. A. , "Making Computers Think Like People," *IEEE Spectrum*, vol. 8, pp: 26-32, 1984.

[31] Simpson, P. K, "Fuzzy min-max neural networks -- Part 2: Clustering," *IEEE Transactions on Fuzzy Systems,* vol.1:1, pp: 32-45. Feb. 1993.

[32] Bezdek, J.C. "Computing with uncertainty", IEEE Communications Magazine, vol: 30:9, pp: 24:36, Sept. 1992.

[33] Kosko, B., Neural Networks and Fuzzy Systems: A Dynamic System Approach to Machine Intelligence, Prentice-Hall, London, 1992

[34] Lefteri H., Tsoukala ,Robert E. Uhrig, "Fuzzy and Neural Approaches in Engineering," New York: John wily &sons, Inc. 1997.

[35] Kwan H. K., Cai Y., "Fuzzy Neural Network and Its Application to Pattern Recognition," *IEEE Transaction on Fuzzy Systems*, vol.2, No. 3, pp: 185-193,1994.

[36] Hirota K., Pedrycz W., "OR/AND Neuron in Modeling Fuzzy Set Connectives," *IEEE Transactions on Fuzzy Systems*, vol. 2:2, pp: 151-161, May 1994.

[37] Pedrycz W., Rocha A., " Fuzzy-set based models of neurons and knowledge-based networks," *IEEE Transactions on Fuzzy Systems*, vol:1:4, pp: 254 –266, Nov. 1993.

[38] Pfluger N., Yen J., Langari R.,"A defuzzification strategy for a fuzzy logic controller employing prohibitive information in command formulation," *IEEE International Conference on Fuzzy Systems*, pp: 717-723, 8-12 Mar 1992.

[39] Halgamuge S.K., Runkler T.A., Glesner M.,"On the neural defuzzification methods," *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, vol.1, pp: 463-469, Sept. 1996.

[40] Lee C.C., "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II," *IEEE Transactions on Systems, Man and Cybernetics*, vol.20(2), pp: 419-435, 1990.

[41] Kang G., Lee W., Sugeno M.," Design of TSK fuzzy controller based on TSK fuzzy model using pole placement," *The 1998 IEEE International Conference on Computational Intelligence*, vol.1, pp: 246-251, May 1998.

[42] Sugeno M., "On stability of fuzzy systems expressed by fuzzy rules with singleton consequents," *IEEE Transactions on Fuzzy Systems*, vol.7. Issue: 2, pp: 201-224,Apr. 1999.

[43] Castellano G., Fanelli A. M., Pelillo M., "Pruning in recurrent neural networks", *Proceedings of International Conference on Artificial Neural Networks (ICANN'94)*, Sorrento, Italy, pp: 451–454, 1994.

[44] Castellano G., Fanelli A.M., Pelillo M., "Iterative pruning in second- order recurrent neural networks," *Neural Processing Letters*, vol. 2, No. 6, pp: 1–4, 1995.

[45] Abu-al-nadi D., Popovic D., " Texture analysis using an adaptive neuro-fuzzy network and fractals," Proceedings of IEEE International Conference on Fuzzy Systems FUZZ-IEEE '99, vol.2, pp: 1102-1107, 1999.

[46] Guely F., Siarry P., " Gradient Descent Method for Optimizing Various Fuzzy Rule Bases," Second IEEE International Conference on Fuzzy Systems, vol.2, pp: 1241 -1246, 1993.

[47] Kothari R., "Robust Regression Based Training of ANFIS," 18th International Conference of the North American Fuzzy Information Processing Society, pp: 605 –609, 1999.

[48] Chiu S. L., "Fuzzy Model Identification Based on Cluster estimation," *Journal of Intelligent Computing and Fuzzy Systems,* vol. 2 (3), 1994.

[49] Krishna P. R., Kanal L. N., "Classification, Pattern recognition, and reduction of dimensionality, vol.2 of Handbook of Statistics, North-Holland, Amsterdam, 1982.

[50] Makhoul J., Roucos S., Gish H., "Vector Quantization in Speech Coding," Proceedings of IEEE, vol. 73(11): pp:1551-1588,1985.

[51] Mehrubeoglu M., Kehtarnavaz N., Marquez G., Wang L.V. "Characterization of Skin lesion Texture in Diffuse Reflectance Spectroscopic Images," Image Analysis and Interpretation, 2000. Proceedings. 4th IEEE Southwest Symposium, pp: 146-50, 2000.

[52] Krattenthaler W., Mayer K. J., Zeiller M., "Point correlation: A reduced cost template matching technique," Proc. IEEE Int. Conference of Image Processing, vol. 1, Austin, TX, pp: 208–212, Nov. 13–16, 1994

[53] Rosenfeld A., Kak A., *Digital Image Processing*, 2nd ed. New York: Academic, 1982, vol. 2.

[54] Jain A. K., *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[55] Thanaka K., Sano M., Ohara S., Okudaira M., "A parametric Template Method and its Application to Robust Matching," Proceedings. IEEE Conference on Computer Vision and Pattern Recognition *2000*, vol.1, pp: 620 –627, 2000.

[56] Ullman S., Basri R., "Recognition by linear combinations of models, " *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol:13 Issue: 10 , pp: 992-1006,Oct 1991.

[57] Jain, A.K., Zhong Y., Lakshmanan S., "Object matching using deformable templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol: 18, Issue: 3,pp: 267 –278, Mar 1996.

[58] Jurie F., Dhome M. "A simple and efficient template matching algorithm," Proc. of Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol.2, pp: 544-549, 2001.

[59] Hornik K., Stinchcombe M., White H., "Multi Layer Feedforward Networks are universal approximators," *Neural Networks*, vol. 2, pp. 559–366, 1989.

[60] Thananchai L., "Novel Determination of Differential-Equation Solutions: Universal Approximation Method," *Journal of Computational and Applied Mathematics*, vol.146, pp: 443–457, 2002.

[61] Scarselli F., Ah Chung T., "Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results," *Neural Networks*, vol. 11, No.1, pp: 15-37, 1998.

[62] Villiers J. d., Barnard E., "Backpropagation neural nets with one and two hidden layers," *IEEE Transactions on Neural Networks*, vol. 4(1), pp:136–141, 1992.

[63] Cybenko G., "Approximation by Superpositions of Sigmoidal Functions," *Mathematics of Control, Signal, and System*, vol. 2,pp: 303-314,1989.

VITA AUCTORIS

Idris S. El-Feghi

Idris S. El-Feghi was born on March 28th, 1959, in Misurata-Libya. He received his B.Sc degree in Computer Engineering form University of Portland, Oregon, USA in 1983. He worked as a computer engineer in Tajura Research Center from 1983-1988.

In Nov. 1988 he worked for the General Secretariat of Industry, first as a system administrator and then as the information center manager during the time period of 1990-1993.

From 1993-1995 he worked in Misurata High Polytechnic Institute as a computer instructor. He taught a number of courses in computer design and programming.

In 1999 He received his M.A.Sc in Electrical Engineering from university of Windsor, located in Windsor, Ontario, Canada.

In Sept. 1999 he started his Ph.D program in the area of biomedical image analysis at University of Windsor. He has taught several senior level courses in the Electrical and Computer Engineering Department in addition to holding research and teaching assistantships in the same department. His current areas of research include, Machine Vision, Neural Networks, Pattern Recognition and Image Processing.

Currently he is a candidate for Ph.D degree in Electrical and Computer Engineering from University of Windsor, Windsor, Ontario, Canada.

Publications:

Referred Journal Publications

[1] I.El-Feghi, M.Ahmadi, W.C. Miller, M. A. Sid-Ahmed and V. Ramachandran "Design of Stable 3D Recursive Digital Filters Using 3-Variables Very Strictly Hurwitz Polynomial," *Journal of Circuits, systems and Computers*, Vol. 10,Nos.5 &6 (2000) pp 219-228.

[2] I.El-Feghi, M. A. Sid-Ahmed and M.Ahmadi "Automatic Localization of Craniofacial Landmarks for Assisted Cephalometry," *Journal of Pattern Recognition* (to appear)

[3] I.El-Feghi, M.A.Sid-Ahmed and M.Ahmadi "Identification and Localization of Cephalometric Landmarks Using Multi Layer Perceptron," *IEEE Transaction of Biomedical Engineering (submitted)*

Invited Papers

[1] I. El-Feghi, M.A.Sid-Ahmed and M.Ahmadi "Automatic Localization of Craniofacial Landmarks for Assisted Cephalometry " *IEEE International Symposium on Circuits and Systems* (ISCAS) 2003, Bangkok, Thailand, May 25-28, 2003

Conference Proceedings Publications

[1] I. El-Feghi, M.A.Sid-Ahmed and M.Ahmadi, "Design of 3-D Recursive Digital Filters Using Linear Programming," 5th *ISSPA* 1999, Brisbane, Australia, PP. 93-976, Aug.1999.

[2] I. El-Feghi, M. Ahmadi, M. A. Sid-Ahmed, "Design of 3-D Recursive Digital Filters With Separable Denominator and Non-Separable Numerator," *Third International Symposium on Intelligent Automation and Control,* Manu, Hawaii. Jun 11-16, 2000

[3] I. El-Feghi, M. Ahmadi, W.C. Miller, V. Ramachandran and M.A. Sid-Ahmed, "A Method For Generation Of 3-Variable VSHP and Its Application in The Design Of 3-Dimensional Digital Filters", *Third International Symposium on Intelligent Automation and Control,* Maui, Hawaii, June 11-16, 2000

[4] I. El-Feghi, M. A. Sid-Ahmed and M. Ahmadi, "Location of Craniofacial Landmarks on X-ray Images by Employing Fuzzy Neural Networks," 45th *Midwest Symposium on Circuits and Systems*, Tulsa, Oklahoma USA, vol.3, pp 348-351, August 2002

[5] I. El-Feghi, M. A. Sid-Ahmed and M. Ahmadi, "Improving Cephalometric Landmark Localization using Multi-Layer Perceptron, " *ICISP'2003*, Agadir, Morocco, June 25-27, 2003

[6] I. El-Feghi, M. A. Sid-Ahmed and M. Ahmadi, "Identification and Localization of Cephalometric Landmarks using Multi Layered Neural Network," NEWCAS, Montreal, Canada, June 17-20, 2003

[7] I. El-Feghi, M. A. Sid-Ahmed and M. Ahmadi "Automatic Identification and Recognition of Craniofacial Landmarks using Neural Networks Approach," *Medical Image Computing & Computer Assisted Intervention*, Toronto, Canada, November 1-4, 2003